Università
degli Studi
di Padova

Dipartimento
di Ingegneria
dell'Informazione

Multi-Agent Systems in Smart Environments

from sensor networks to aerial platform formations

Ph.D. candidate
Giulia Michieletto

Advisor
prof. Angelo Cenedese

Director & Coordinator
prof. Andrea Neviani

Ph.D. School in
Information Engineering

Department of
Information Engineering
University of Padova
2017

Multi-Agent Systems in Smart Environments

from sensor networks to aerial platform formations

Ph.D. candidate: Giulia Michieletto

Director: prof. Andrea Neviani
Advisor: prof. Angelo Cenedese

Ph.D. School in Information Engineering
Department of Information Engineering
University of Padova
2017

“I was originally supposed to become an engineer but the thought of having to expend my creative energy on things that make practical everyday life even more refined, with a loathsome capital gain as the goal, was unbearable to me.”
Albert Einstein

To you, who have always encouraged me to be myself.

Abstract

In the last twenty years, the advancements in pervasive computing and ambient intelligence have lead to a fast development of smart environments, where various cyber-physical systems are required to interact for the purpose of improving human life. The effectiveness of a smart environment rests thus upon the cooperation of multiple entities under the constraints of real-time high-level performance. In this perspective the role of multi-agent systems is evident due to capability of these architectures involving large sets of interactive devices to solve complex tasks by exploiting local computation and communication. Although all the multi-agent systems arise for scalability, robustness and autonomicity, these networked architectures can be distinguished according to the characteristics of the composing elements. In this thesis, three kinds of multi-agent systems are taken into account and for each of them innovative distributed solutions are proposed to solve typical issues related to smart environments.

Wireless Sensor Networks - The first part of the thesis is focused on the development of effective clustering strategies for wireless sensor network deployed in industrial environment. Accounting for both data clustering and network decomposition, a centralized and a distributed algorithms are proposed for grouping nodes into local non-overlapping clusters in order to enhance the network self-organization capabilities.

Multi-Camera Systems - The second part of the thesis deals with the surveillance task for networks of interoperating smart visual sensors. First, the attitude estimation step is handled facing the determination of the orientation of each device in the group with respect to a global inertial frame. Afterwards, the perimeter patrolling problem is addressed, within the border of a certain area is required to be repeatedly monitored by a set of planar cameras. Both issues are recast in the distributed optimization framework and solved through the iterative minimization of a suitable cost function.

Aerial Platform Formations - The third part of the thesis is devoted to the autonomous aerial platforms. Focusing on a single vehicle, two desirable properties are investigated, namely the possibility to independently control the position and the attitude and the robustness to the loss of a motor. Two non-linear controllers are then designed to maintain a platform in static hovering keeping constant reference position with constant attitude. Finally, the interest is moved to swarms of aerial platforms aiming at both stabilizing a given formation and steering it along pre-definite directions. For this purpose, the bearing rigidity theory is studied for frameworks embedded in the three-dimensional Special Euclidean space.

The thesis thus evolves from fixed to fully actuated multi-agent systems accounting for smart environments applications dealing with an increasing number of DoFs.

Sommario

Nell’ultimo ventennio, i progressi nel campo della computazione pervasiva e dell’intelligenza ambientale hanno portato ad un rapido sviluppo di ambienti smart, dove più sistemi cyber-fisici sono chiamati ad interagire al fine di migliorare la vita umana. L’efficacia di un ambiente smart si basa pertanto sulla collaborazione di diverse entità vincolate a fornire prestazioni di alto livello in tempo reale. In quest’ottica, il ruolo dei sistemi multi-agente è evidente grazie alla capacità di queste architetture, che coinvolgono gruppi di dispositivi capaci di interagire tra loro, di risolvere compiti complessi sfruttando calcoli e comunicazioni locali. Sebbene tutti i sistemi multi-agenti si caratterizzino per scalabilità, robustezza e autonomia, queste architetture possono essere distinte in base alle proprietà degli elementi che le compongono. In questa tesi si considerano tre tipi di sistemi multi-agenti e per ciascuno di questi sono proposte soluzioni distribuite e innovative volte a risolvere problemi tipici per gli ambienti smart.

Reti di Sensori Wireless - La prima parte della tesi è incentrata sullo sviluppo di efficaci strategie di clustering per le reti di sensori wireless impiegate in ambito industriale. Tenendo conto sia dei dati acquisiti che della topologia di rete, sono proposti due algoritmi (uno centralizzato e uno distribuito) volti a raggruppare i nodi in clusters locali non sovrapposti per migliorare le capacità di auto-organizzazione del sistema.

Sistemi Multi-Camera - La seconda parte della tesi affronta il problema di videosorveglianza nel contesto di reti di sensori visivi intelligenti. In primo luogo, è considerata la stima di assetto che prevede la ricostruzione dell’orientamento di ogni agente appartenente al sistema rispetto ad un sistema globale inerziale. In seguito, è affrontato il problema di pattugliamento perimetrale, secondo il quale i confini di una certa area devono essere ripetutamente monitorati da un insieme di videocamere. Entrambe le problematiche sono trattate nell’ambito dell’ottimizzazione distribuita e risolte attraverso la minimizzazione iterativa di un’adeguata funzione costo.

Formazioni di Piattaforme Aeree - La terza parte della tesi è dedicata alle piattaforme aeree autonome. Concentrandosi sul singolo veicolo, sono valutate due proprietà, ovvero la capacità di controllare indipendentemente la posizione e l’assetto e la robustezza rispetto alla perdita di un motore. Sono quindi descritti due controllori non lineari che mirano a mantenere una data piattaforma in hovering statico in posizione fissa con orientamento costante. Infine, l’attenzione è volta agli stormi di piattaforme aeree, studiando sia la stabilizzazione di una determinata formazione che il controllo del movimento lungo direzioni prefissate. A tal fine viene studiata la teoria della bearing rigidità per i sistemi che evolvono nello spazio speciale euclideo tri-dimensionale.

La tesi evolve dunque dallo studio di sistemi multi-agenti fissi a totalmente attuati usati in applicazioni per ambienti smart in cui il numero di gradi di libertà da gestire è incrementale.

Acknowledgments

This thesis represents an important milestone of a journey begun several years ago by a young, doubtful girl with dreamy eyes. The achievement of this goal has been possible because of plenty of people who have accompanied me along this stretch of road and it is therefore compulsory to thank them.

Thanks to my ‘super-advisor’ Angelo Cenedese. To be an enthusiastic, proactive and competent researcher, a polite, honest and understanding man and a friendly, committed and stimulating mentor. Thanks to have invested in me!

Thanks to Antonio Franchi. For having host me in LAAS, for having introduced me in new branches of academic research, for having (often) pushed me beyond my limits.

Thanks to Luca Zaccarian and Hyo-Sung Ahn to have carefully reviewed this thesis providing positive comments and helpful hints.

Thanks to all the people who have worked with me in these years, especially Giuseppe and Markus.

Thanks to all the (past and current) inhabitants of the offices 329, 330 and 326. Especially, to Andrea, Marco, Diego, Giacomo, Nicoletta, Irene and Chen for the scientific discussions and the frivolous talks. Thanks to Michele, with whom I have shared most of my academic career, for having shared smiles and struggles. Thanks to Chiara and Giulia to be good friends before than co-workers.

Thanks to the guys from Toulouse, in particular thanks to Davide. For having spent with me a lot of great moments inside and outside the lab, discussing about controllability of aerial platforms and about the sense of the universe, climbing and enjoying the city. Merci de tout coeur!

Thanks to the friends ever. To be good companions of joyful evenings and enthusiastic lovers of the pleasures of life. To the boys to let me play the tomboy role, and to the girls to remind me that I am a woman who likes to makeup and style my hair. You are a second family for me!

Thanks to my family, to Sandra e Stefano, to Giacomo. For their immense and constant love, for their unconditional and unfailing support, and for the great confidence in my skills and choices. And to be always proud of what I do without understanding what I really do.

Thanks to those who have taught me to be courageous, to make my choices without fear, to face calmly and rationally the issues without answer. To those who gives me a smile, a hug and a bit of heart every day. To my always and forever. Thanks to Mattia.

Contents

Acronyms

aSST	averaged Single Spanning Tree
aMP	averaged Multipath Method
BC	Bearing Congruent
BE	Bearing Equivalent
BR	Bearing Rigid
BRF	Bearing Roto-Flexible
CCA	Centralized Clustering Algorithm
CH	Cluster Head
CoM	Center of Mass
DCA	Decentralized Clustering Algorithm
DMGA	Distributed Matching-based Grouping Algorithm
DoF	Degree of Freedom
ESC	Electronic Speed Controller
FA	Fully Actuated
FC	Fully Coupled / Failed Controller (according to the context)
FD	Fully Decoupled
FoV	Field of View
GAD	General Anomaly Detection
GBR	Globally Bearing Rigid
GTM	Generically Tilted Multi-rotor
HC	Healthy Controller
ICT	Information and Communication Technology
IMU	Inertial Measurement Unit
IoT	Internet of Things
IWSN	Industrial Wireless Sensor Network
KMC	*k*-Means Clustering
LEACH	Low Energy Adaptive Clustering Hierarchy
MCD	Maximum Common Divisor
MP	Multipath Method
PC	Partially Coupled
PD	Preferential Direction
PP	Preferential Plane
PTZ	Pan-Tilt-Zoom
SPD	Signle Preferential Direction
SPP	Signle Preferential Plane
SST	Single Spanning Tree
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
VSN	Visual Sensor Network
WSN	Wireless Sensor Network

Notation

Symbols

In this manuscript the following notation is adopted. The symbols $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{C}$ respectively denote the set of natural, integer, real and complex numbers, while $\mathbb{R}^+$ ($\mathbb{R}_0^+$) refers to the positive real numbers excluding (including) zero. The symbols $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ indicate the spaces of n-dimensional vectors and $(n\times m)$-dimensional matrices having real entries, respectively. The symbol $\mathbb{S}^n$ identifies the n-dimensional manifold on the unit sphere in $\mathbb{R}^{n+1}$. Lowercase italic letters and greek symbols, e.g., x, v, ϵ, denote scalar values. Lowercase roman bold letters and bold greek symbols, e.g., $\mathbf{x}, \mathbf{v}, \boldsymbol{\epsilon}$, denote vectors assumed by convention to be column vectors. Uppercase roman bold letters, e.g., $\mathbf{X}, \mathbf{A}, \mathbf{C}$, denote matrices. Uppercase calligraphic letters, e.g., $\mathcal{X}, \mathcal{A}, \mathcal{C}$, denote sets. The symbol $\mathbb{1}_n$ represents the n-dimensional (column) vector of all ones. The symbol $\mathbf{I}_n$ denotes the $(n\times n)$-dimensional identity matrix. The symbol $\emptyset$ indicates the empty set.

Operators

Given a scalar x, $|x|$ denotes its absolute value. Analogously, given a n-dimensional vector $\mathbf{v}$, $|\mathbf{v}|$ represents its absolute value, while $\|\mathbf{v}\|_p$ is its p-norm. The (row) vector $\mathbf{v}^\top$ indicates the transpose of $\mathbf{v}$ and the scalar value v_i specifies its i-th entry, i.e., $v_i = [\mathbf{v}]_i$. For the vectors, the symbols $>, \geq, =, \leq, <$ indicate component-wise relations. In addition, $\mathrm{diag}(\mathbf{v})$ denotes the $(n\times n)$ diagonal matrix whose main diagonal consists of the elements of $\mathbf{v}$. Assuming that $\mathbf{v} \in \mathbb{R}^3$, $[\mathbf{v}]_\times$ is its corresponding (3×3) skew-symmetric matrix belonging to the Lie algebra $\mathfrak{so}(3)$. Denoting with $\times$ the cross product between two vectors $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^3$, it holds that $\mathbf{u} \times \mathbf{v} = [\mathbf{u}]_\times \mathbf{v}$.

Given a matrix $\mathbf{A}$, $\|\mathbf{A}\|_p$ and $\mathbf{A}^\top$ denote the same operations of norm and transpose as before, while the operators $\mathrm{rk}(\mathbf{A})$, $\mathrm{tr}(\mathbf{A})$, $\det(\mathbf{A})$, $\mathrm{Im}(\mathbf{A})$ and $\ker(\mathbf{A})$ indicate the matrix rank, trace, determinant, image space and null space, respectively. In addition, it is $\mathbf{A} \geq 0$ ($\mathbf{A} > 0$) when the matrix $\mathbf{A}$ is positive semidefinite (definite) and the symbol $\leq$ ($<$) is used for negative semidefinite (definite) case. The symbol a_{ij} indicates the ij-th scalar entry of the matrix, namely $a_{ij} = [\mathbf{A}]_{ij}$. A matrix $\mathbf{A}$ is said to be non-negative if its entries are all non-negative. If $\mathbf{A}$ is a generic matrix, it is always possible to compute its pseudoinverse $\mathbf{A}^\dagger$, whereas if it is a square matrix, $\mathbf{A}^{-1}$ specifies its inverse and $\Lambda(\mathbf{A})$ denotes the set containing its eigenvalues. A $(n\times n)$ matrix $\mathbf{A}$ is said to be stochastic if it is non-negative and satisfies $\mathbf{A}\mathbb{1}_n = 1$. Moreover, it is said doubly stochastic if it is stochastic and $\mathbb{1}_n^\top \mathbf{A}\mathbb{1}_n = 1$. A stochastic matrix $\mathbf{A}$ is primitive if it has only one eigenvalue equal to 1 and all the other eigenvalues are strictly inside the unitary circle.

Given a set $\mathcal{C}$, the symbol $|\mathcal{C}|$ denotes its cardinality, i.e., the number of elements belonging to the set. Given two sets $\mathcal{A}$ and $\mathcal{C}$, the relation $\mathcal{C} \subset \mathcal{A}$ ($\mathcal{C} \subseteq \mathcal{A}$) indicates that $\mathcal{C}$ is a proper (non necessarily proper) subset of $\mathcal{A}$. The symbols $\cap$, $\cup$ and $\setminus$ represent intersection, union and complement of sets, respectively. In addition, in this case, the symbol $\times$ denotes the Cartesian product, while $+$ denotes the (not directed) sum.

1

Introduction

1.1 Towards Smarter Environments

In last decades the concept of *smart environments* has gained a lot of popularity thanks to the rapid advancement of both the digital and robotic technologies and the research in the field of machine learning, distributed optimization and automatic controls.

The definition of smart environments evolves from the concept of ubiquitous computing that, according to Weiser, Gold, and Brown (1999), promotes the ideas of 'a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network'. Cook and Das (2004) have described a smart environment as 'a small world where different kinds of smart devices are continuously working to make inhabitants lives more comfortable'. From these definitions, it is evident the strong interactive cyber-physical aspect that characterizes an environment denominated *smart*, whose complexity mainly derives from the need of coordination and autonomicity towards the improvement of human life conditions.

According to the given premises, a *smart city* results to be a urban development vision that integrates information and communication technology (ICT) and Internet of things (IoT) technology to improve the economic and political efficiency and to enable the social, cultural and urban development. It is a complex self-monitoring and self-responsing networked infrastructure where human and technological factors are required to harmoniously interact, to improve the citizen life conditions, to optimize the use of the available resources (e.g., water, power, transportation), to act efficiently in critic scenarios. Several examples of efficient smart cities are already present all over the word (see Figure 1.1). In these realities, the adoption of the IoT paradigm is supported by the spread of *smart houses* and *smart buildings*. These technology-oriented residences and buildings are equipped with a network of physical devices, sensors, actuators, and other items embedded with electronics, that, collecting and exchanging data, allows to (remotely) monitor, control and safeguard the inhabitants life (Jiang, Liu, and Yang (2004); Schaffers, Komninos, Pallot, Trousse, Nilsson, and Oliveira (2011); Batty, Axhausen, Giannotti, Pozdnoukhov, Bazzani, Wachowicz, Ouzounis, and Portugali (2012); Zanella, Bui, Castellani, Vangelista, and Zorzi (2014)). In the industrial field, the outburst towards the realization of smarter environments has led to the fourth industrial revolution called Industry 4.0. This is, indeed, related to the emergence of *smart factories*, where the productive processes and the resources management are improved by the employment of interactive cyber-physical systems (Shrouf, Ordieres, and Miragliotta (2014); Hermann, Pentek, and Otto (2016); Wang, Wan, Li, and Zhang (2016)).

Within all the described self-organized scenarios, the *multi-agent systems* technology is unavoidable resting upon structures made up of multiple interacting intelligent units. More specifically, a multi-agent system is a set of interactive elements, able to communicate and cooperate in order to achieve a common goal, e.g., to monitor phenomena of interest, to detect events basing on the (continuous) exchange of information gained by every device displaced in the environment and to control the environment itself (Chong and

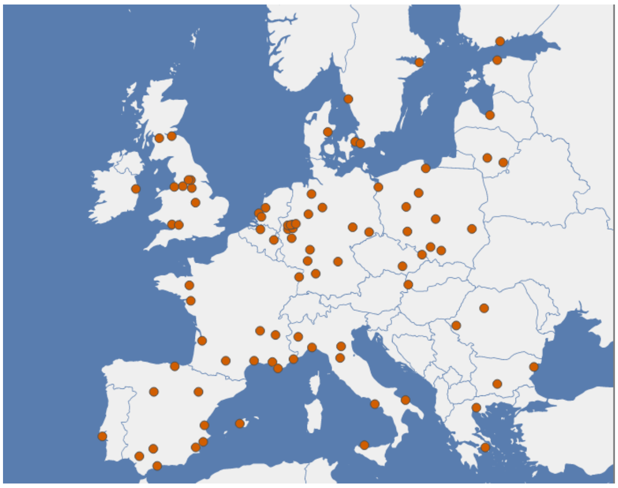


Figure 1.1. European smart cities in 2015 according to the analysis conducted by Vienna University of Technology. [*credits: www.smart-cities.eu*]

Kumar (2003); Hespanha, Naghshtabrizi, and Xu (2007); Wooldridge (2009); Ganzha, Jain, Jarvis, Jarvis, and Rönnquist (2013)). As a consequence, they emerge as a key element for the success of the smart environments, which require to act the development of distributed algorithms that can avoid problems concerning scalability and robustness (of both agents and network) and allow the emergence of collective behaviors through simple local rules implementation (Cook (2009)).

In real-world smart environments, all the gathered measurements and involved models are affected by noise, faults and other nuisances because of their nature and of the human interaction. Therefore, challenging issues for embedded multi-agent systems are the detection, recognition and diversification not only of occurring events, but also of faults that can affect the system, in addition to the identification and characterization of the uncertainty induced by noisy agents in the network, as well as the estimation and management of the noise related to the operating environment itself and to the possible human presence in the control loop. At the same time, a high level of autonomicity is required: a multi-agent system has to act as an autonomous self-aware decentralized intelligence, taking decisions, cooperating and acting on the environment in a safe and non-disruptive way. It is thus necessary for each device to be conscious about its state and to communicate with a set of neighbors for the purpose of updating its knowledge of the network and making progress in the task execution. Hence, ubiquitous issues in multi-agent systems are localization, which results in determining the self and neighbors position and orientation, and the management of the communication under the constraints of reliability and real-time performance.

Motivated by the desire of developing even smarter environments, the design and control of efficient interactive multi-agent systems remains an interesting research area that offers several study directions. In particular, in this thesis, various kinds of multi-devices structures are taken into account; for each of them the attention is focused on traditional and innovative problems which can arise in the smart environments for the purpose of proposing new and improved solutions.

1.2 Networked Multi-Agent Systems

A multi-agent system can be described as a set of autonomous entities that interact among them to solve problems that are beyond the capacity and knowledge of each individual agent (Potiron, Seghrouchni, and Taillibert (2013)). The strength of this kind of architecture is indeed the possibility to accomplish complex global tasks through the realization of simple local rules from interactive agents. A multi-agent system is thus mainly characterized by a cooperative nature which is not altered by the addition and/or removal of a subset of agents. In this sense, it is robust and scalable: robustness refers to the ability of a system to tolerate the failure of one or more agents, while scalability originates from the system modularity.

Although a multi-agent system can be composed of different kinds of devices constituting an heterogeneous system in which different capabilities can be combined in a convenient way, in this thesis the attention is focused on *homogeneous* systems where all the agents share the same nature. In detail, each element of such a multi-device architecture is an autonomous and controllable entity characterized by the features summarized in the next sections.

1.2.1 Agent Properties

Location

A multi-agent systems acts in a certain environment of interest, that can be a two-dimensional or three-dimensional space according to the considered task and the imposed constraints. In this scenario, each element of the multi-device group is characterized by a (fixed or time-varying) location or pose whose definition depends on the agent model adopted. In the following, two possible models are assumed: *massless point* or *rigid body*.

In the first case, the physical features (mass and inertia) of the agent are neglected, thus its location in the space of interest is described only by its position in the global inertial frame. In other words, its pose is defined by the position vector $\mathbf{p} \in \mathbb{R}^d$, where $d \in \{2, 3\}$ is the dimension of the considered space.

On the contrary, when the agent is modeled as a rigid body, it is necessary to consider the body frame attached to the device so that the origin generally coincides with the agent center of mass (CoM). Hence, the location of the agent in the space is given by the pair constituted by the position of its CoM (namely, the position of the origin of its body frame) in the global inertial world frame and by the orientation (or attitude) of the body frame with respect to the inertial world frame. Formally, the device pose is defined by $(\mathbf{p}, \mathbf{R}) \in \mathbb{R}^d \times SO(d)$, where $d \in \{2, 3\}$ and $\mathbf{R}$ is a $(d \times d)$ rotation matrix[1] belonging to the Special Orthogonal group $SO(d) = \{\mathbf{R} \in \mathbb{R}^{d \times d} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}_d, \det(\mathbf{R}) = +1\}$ that describes the rigid bodies orientation in the d-dimensional space.

[1] In some sections of this manuscript, another formalism will be adopted to represents the attitude of the considered agents, namely the unit quaternions.

Motion Capability

Several multi-agent systems applications require the autonomous exploration of a certain domain of interest. This is the case of search-and-rescue operations, wide areas aerial surveillance, human-inaccessible environments mapping. In these scenarios, the employed devices must has a certain level of mobility. On the other hand, global activities as the persistent monitoring of the (indoor/outdoor) environment for the purpose of detecting fault and anomalies could be performed even using a network of fixed devices.

Given these premises, in this thesis, three categories are distinguished as regards for motion capability of each agent in a multi-device group, affecting its controllable DoFs, namely

- *no motion capability*: the agent is fixed, hence it has zero DoFs;
- *partial motion capability*: the DoFs of the agent are less than the number of parameters that describe its location, thus it could have fixed position and variable orientation or viceversa or as well its translational/rotational motion could be constrained on certain movements only;
- *full motion capability*: the agent can vary both its position and orientation over time, so its DoFs coincide with the number of parameters required to depict its pose in the space.

Sensing Capability

Each agent is assumed to be provided with sensing capability, namely to be able to gather information about the neighboring agents in the system and/or on the surrounding environment. The achieved measurements depend on the nature of the device itself and/or of the sensors with which it is equipped both in terms of typology and accuracy.

Most used sensors can be classified into five main classes: (*i*) *environmental* sensors, which retrieve information about a specific quantity (e.g., temperature, humidity, pressure, light) from the environment; (*ii*) *proximity* sensors, which reveal the presence of something (e.g., other agents, obstacles) around the agent; (*iii*) *visual* sensors, which capture frames of a certain scene in a continuous or sampled manner; (*iv*) *range* sensors that provide measurements of distance; and (*v*) *bearing* sensors that sense the relative direction.

An RGB camera, for example, could be considered both as an agent in a large scale surveillance system and as a visual sensor assembled on a vehicle; on the other hand, a team of ground robots for terrestrial exploration is a set of agents, each of them generally equipped with multiple sensors to allow the autonomous navigation.

Communication Capability

Because of the interactive nature of the multi-agent systems, each element of the group must have communication capability, i.e., it has to be able to propagate information to others devices in the system. In real-world scenarios, the communication range of a device is generally constrained, hence it is usually assumed that each agent can interact limited to the neighbors in a certain area.

In addition, the information exchange in a given system is generally governed by a predetermined communication protocol that regulates the transmission/reception of the messages and the successive update of the agents state. The most used communication strategies are the following:

1. synchronous,
2. asynchronous broadcast,
3. asynchronous symmetric gossip,
4. asynchronous asymmetric gossip.

In the first case all the agents in the group communicate their information to all their neighbors at the same instant and then they all update their state. In the second case, at each interaction only one device communicates with its neighbors and only its neighbors update their states. In the third case, at each interaction, only a pair of agents exchanges information and only these two agents update their states. Finally, the fourth protocol is similar to the previous one but in this case communication is not bidirectional as in the symmetric case, namely only one agent sends information to the other one (which updates its state) and not viceversa.

Computational Capability

In smart environments context, the trend is to avoid as much as possible the presence of a single high powerful control unit, which elaborates all the device data in order to produce the required results. It is instead preferred the adoption of a decentralized paradigm, according to which each agent is required to store and manage a certain amount of information. For this reason, depending on the application task, the devices constituting a multi-agent system are generally equipped with a microprocessor with a suitable memory buffer and the computational resources necessary to provide a local contribution to the distributed task solution.

1.2.2 Graph-Based Representation

According to the sensing and/or communication capabilities, a multi-agent system made up of n devices can always be represented with a network composed by n nodes organized into a (directed/undirected) graph (Mesbahi and Egerstedt (2010)).

Formally, a *graph* $\mathcal{G}$ is an ordered pair $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1 \dots v_n\}$ is the set of nodes/vertices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. It is assumed that the vertices of the graph correspond to the elements of the network, namely to the devices in the multi-agent system, so that $|\mathcal{V}| = n$. In addition, it is assumed that the pair (v_i, v_j) belongs to the set $\mathcal{E}$ if and only if the node/agent v_i can communicate with the node/agent v_j or 'sense' the

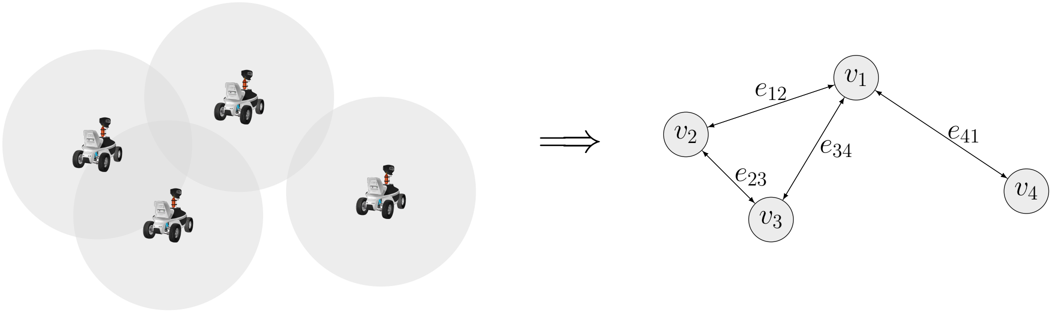


Figure 1.2. Example of a multi-robot system represented by an undirected graph. Each agent in the group is characterized by a finite communication/sensing radius (gray disc), for each pair of interacting robots there exists an edge in the corresponding graph.

node/agent v_j being able to retrieve some relative measurements. The first case refers to the *communication graph* of the given system, while in the second case the *sensing graph* is taken into account.

For sake of simplicity, in this thesis, as often in literature, sensing and communication graphs are always supposed to be coincident, nevertheless the cases of bilateral and unilateral agent interactions are distinguished. When the communicating devices reciprocally exchange information, i.e., when $(v_i, v_j) \in \mathcal{E}$ if and only if $(v_j, v_i) \in \mathcal{E}$, then the corresponding graph is said to be *undirected* or *non-oriented* (see Figure 1.2 for an example of multi-robot system associated to an undirected graph). Contrarily, the underlying graph is called *directed* or *oriented* when $(v_i, v_j) \in \mathcal{E}$ does not imply $(v_j, v_i) \in \mathcal{E}$. In this case, the neighboring agents interaction is not necessary mutual and the edge $(v_i, v_j) \in \mathcal{E}$ is directed from v_i to v_j meaning that the i-th agent can detect the j-th agent but not viceversa.

Independently on the nature of the interplays, when all the devices in the group can interact among them (e.g., ideally assuming infinite communication/sensing radius), the related graph is named *complete*: for each pair of vertices there exists an edge that connects them, i.e., $(v_i, v_j) \in \mathcal{E}\ \forall (v_i, v_j) \in \mathcal{V} \times \mathcal{V}$. Denoting with m the cardinality of the edge set ($m = |\mathcal{E}|$), it is straightforward that $m = n(n-1)/2$ for an undirected complete graph and $m = n(n-1)$ for a directed complete graph. In the rest of the thesis, the (undirected/directed) complete graph is always referred as $\mathcal{K}$.

The relationship between the sets $\mathcal{V}$ and $\mathcal{E}$ of a given graph $\mathcal{G}$ is usually described by the *incidence matrix* $\mathbf{E} \in \mathbb{R}^{n \times m}$. For a undirected graph, this is a $\{0, 1\}$-matrix such that $e_{ik} = 1$ if the i-th vertex is incident to the k-th edge (having fixed an arbitrary labelling on $\mathcal{V}$ and $\mathcal{E}$). For a directed graph, the incidence matrix distinguishes between the outgoing and ingoing edges of each node, so that

$$e_{ik} = \begin{cases} 1 & \text{if} \quad e_k = (v_i, v_j) \in \mathcal{E} \quad \text{(outgoing edge)} \\ -1 & \text{if} \quad e_k = (v_j, v_i) \in \mathcal{E} \quad \text{(ingoing edge)} \\ 0 & \text{otherwise} \end{cases}$$

The interconnections among the nodes of the graph are instead generally represented by the *adjacency matrix* $\mathbf{A} \in \mathbb{R}^{n\times n}$ such that $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Note that the adjacency matrix results to be symmetric if $\mathcal{G}$ is undirected.

For each node v_i in $\mathcal{G}$, the set $\mathcal{N}_i = \{v_j \in \mathcal{V} \mid a_{ij} = 1\}$, named *neighborhood*, represents the set of agents interacting with the i-th agent. By convention, it holds that $v_i \notin \mathcal{N}_i$. The cardinality of $\mathcal{N}_i$ is the *degree*, $\deg(v_i)$, of the i-th agent, which corresponds to the i-th element in the main diagonal of the *degree matrix* $\mathbf{D} = \mathrm{diag}(\mathbf{A}\mathbb{1}_n) \in \mathbb{R}^{n\times n}$. The matrix $\mathbf{D}$ in turn contributes to the definition of the *Laplacian matrix*, $\mathbf{L} = \mathbf{D} - \mathbf{A} \in \mathbb{R}^{n\times n}$, whose spectral properties are linked to the connectivity of the graph.

Given a graph $\mathcal{G}$, a *path* ℓ_{ij} from v_i to v_j consists of the nodes sequence $\ell_{ij} = \{v_i \ldots v_j\}$, where for each pair of consecutive elements v_k and v_{k+1} there exists the edge $(v_k, v_{k+1}) \in \mathcal{E}$. A path ℓ_{ij} is said to be simple if all the nodes in the sequence are distinct, while it is called *cycle* when $v_i = v_j$, namely the nodes sequence stars and ends with the same vertex. A graph $\mathcal{G}$ without cycles is said to be *acyclic*, while it is *connected* if there is at least a path for any pair of nodes in the network, i.e., there exists $\ell_{ij}\ \forall (v_i, v_j) \in \mathcal{V} \times \mathcal{V}$. To guarantee this condition, it is necessary and sufficient that the Laplacian matrix of the graph has rank $n - 1$ or equivalently its second smaller eigenvalue (called *algebraic connectivity* or *Fiedler value*) is positive. Given a connected graph $\mathcal{G}$, it is always possible to determine a *rooted spanning tree*, namely a connected and acyclic subgraph with the same vertex set of $\mathcal{G}$ and a selected node r, called *root*. Formally, this can be defined as $\mathcal{ST}_{\mathcal{G},r} = \{(\mathcal{V}', \mathcal{E}') \,|\, \mathcal{V}' = \mathcal{V},\, \mathcal{E}' \subset \mathcal{E} \text{ s.t. } \exists\ \ell_{r,i}\, \forall v_i \in \mathcal{V}, v_i \neq v_r\}$.

To conclude, two particular kinds of graph are here introduced as they will be considered in this thesis: the circulant and the random geometric graphs. A *circulant graph* is an undirected graph having a cyclic group of symmetries, namely where each node v_i is adjacent to the $(i + j)$-th and $(i - j)$-th nodes for each j in a certain list. A *random geometric graph* is an undirected graph constructed by randomly placing the nodes in a unit square and linking two of them if their Euclidean distance is smaller than $r = 0.85\sqrt{\frac{2 \log n}{n}}$, to ensure the graph is connected with high probability (Gupta and Kumar (2000)).

1.3 Thesis Contribution

In this thesis, different types of (homogeneous) multi-agent systems are analized seeking for efficient distributed solutions to face several typical and new issues within the smart environment context. To describe the followed research direction, it is introduced here an *auxiliary scheme* which is meant to improve the reader comprehension through the whole manuscript.

Reported in Figure 1.3, this is a three-dimensional Cartesian system that aims at summarizing the main features of the devices composing the considered multi-agent network. For this reason, one axis takes care of the agent model distinguishing between massless point and rigid body, a second axis accounts for the dimension of the domain

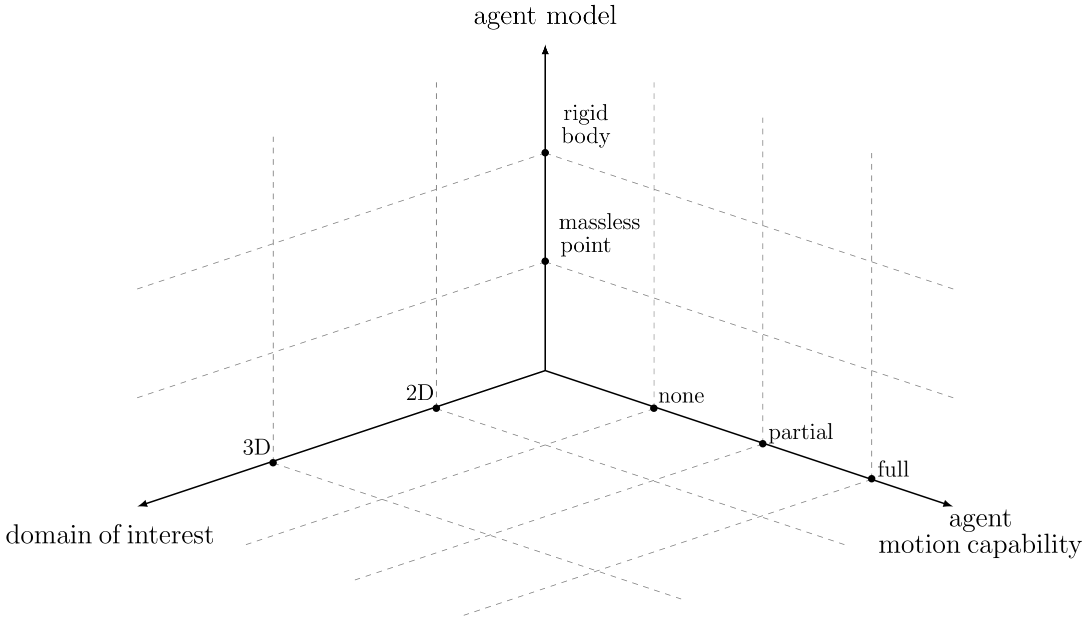


Figure 1.3. Auxiliary scheme to improve the reader comprehension about the research path presented in this thesis.

of interest that can be a two-dimensional or a three-dimensional space, and a third axis considers the agent motion capability according to the categorization provided in Section 1.2.1. Note that the chosen agent model determines the number of parameters necessary to describe the device location.

The outline of the manuscript follows the evolution along the axis of the agent motion capability, starting from the determination of clustering-based fault detection strategies for a set of fixed wireless sensors and arriving to the bearing rigidity based formation control of a swarm of fully-actuated aerial platforms. In this sense, the analysis moves from a static to a dynamic scenario, continuously incrementing the controllable DoFs of every single agent. The main goal of this thesis is thus proposing innovative and robust solutions for estimation and control tasks of different (but homogeneous) multi-agent systems in the light of their coexistence in a real smart environment, where complex problems have to be solved taking care of noise and of possible faults.

1.3.1 Detailed Manuscript Outline

After this chapter where general concepts have been introduced as regards to the multi-agent systems and the smart environments where they intend to act, the rest of the thesis in organized into three main parts.

1. The first part, coincident with Chapter 2, is devoted to the definition of clustering strategies for wireless sensor networks (fixed multi-agent systems) with the intent to design a procedure for detecting failures and anomalies in indoor environments and in industrial production lines.

2. The second part consists of Chapter 3 and Chapter 4 and deals with the surveillance

task for large scale camera networks (partially movable multi-agent systems). For this issue, the knowledge of the location of each agent in the global inertial frame is a key prerequisite; in particular the orientation recovery constitutes the most challenging part because of the reference manifold. To this reason, in Chapter 3 the three-dimensional and two-dimensional attitude estimation is addressed in a decentralized optimization framework. In addition, for the planar camera systems, the perimeter patrolling problem is tackled proposing a suitable distributed strategy in Chapter 4 that relies on the solution of a regularized estimation problem.

3. Chapter 5, Chapter 6 and Chapter 7 make up the third part that copes with the under-actuated and fully-actuated aerial platforms acting in the 3D space (fully movable multi-agent systems). In detail, Chapter 5 provides a preliminary analysis of the fundamental properties of a large class of vehicles, focusing in particular on the robustness with respect to rotor losses. The concept of robustness rests upon the capability of the platform to fly in a constant spot after the damage. The static hovering capabilities are considered also in Chapter 6 where two different and ad-hoc controllers are presented. The interest towards static hovering is motivated by the fact that it is a suitable flight condition for various applications and because it represents a sort of minimum requirement to ensure a gentle emergency landing in case of propeller failure. Finally, Chapter 7 takes into account a richer scenario constituted by a swarm of flying vehicles required to move along specific trajectories while keeping a desired formation. A bearing rigidity based controller is proposed to solve this task.

The manuscript ends with the conclusive Chapter 8 that summarizes the principal achieved results in the studied estimation and control tasks and points out possible future research directions. Finally, Appendix A is devoted to the supplementary material which integrates the work described in the previous chapters, whereas Appendix B provides some auxiliary notions about the mathematics and geometry of the special manifolds considered in the thesis and about the quaternion formalism for the 3D attitude description.

2

Clustering Strategies for WSNs

Supported by the technological advances in creating cheap and compact devices, large scale sensor networks have nowadays become a mature technology for a wide range of application fields. Specifically in industrial automation scenario, the exploitation of wireless multi-sensor systems has shown to provide numerous benefits. By removing the cable infrastructure, the wireless architecture enables the possibility for nodes in a network to dynamically and autonomously group into clusters according to the communication features and the data they collect. This capability allows to leverage the flexibility and robustness of industrial wireless sensor networks in supervisory intelligent systems for high level tasks.

In this chapter, a clustering strategy is studied that partitions a sensor network into a non-fixed number of non-overlapping clusters according to the communication network topology and the measurements distribution: to this aim, both a centralized and a distributed algorithms are designed that do not require a cluster-head structure or other topological assumptions. The developed decentralized clustering procedure is then exploited to autonomously detect anomalies and failure within the smart factory context.

The contents of this chapter are available in
Bianchin G., Cenedese A., Luvisotto M., and Michieletto G. Distributed fault detection in sensor networks via clustering and consensus. *IEEE 54th Annual Conference on Decision and Control (CDC)*, pages 3828–3833, 2015;
Cenedese A., Luvisotto M., and Michieletto G. Distributed clustering strategies in industrial wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 13(1): 228–237, 2017.

2.1 Wireless Sensor Networks

A Wireless Sensor Network (WSN) is a multi-agent system composed of a large number of tiny cheap low-power devices, distributed over a vast area, linked by a wireless medium and equipped with limited computational capabilities (Hespanha et al. (2007)). In early past years, the definition of algorithms and models to manage such networked architectures has emerged among the priority targets of research. At the same time, the application domains of large scale sensor networks are growing and currently range from factory automation to ambient assisted living (Chong and Kumar (2003); Akyildiz and Kasimoglu (2004); Romer and Mattern (2004)).

A WSN can be composed by tens, hundreds or even thousands sensor nodes (also known as *motes*). Each of them arises from the assembly of several highly miniaturized and engineered components. Typically they include a radio transceiver with an internal antenna, a microcontroller unit, an embedded power source and an electronic circuit for interfacing with the sensors. The size and the cost of the sensor nodes is variable: they can measure from a few tenths of a millimetres to 20-30 centimetres and cost from a few to hundreds of dollars, depending on their complexity and performance (Sohraby, Minoli, and Znati (2007); Dargie and Poellabauer (2010)).

While the characteristics of the devices employed in the specific applications may be quite different, all the WSNs are devised according to common criteria and employ collaborative strategies in order to ensure remarkable features such as scalability, adaptability and robustness (Karl and Willig (2007)). To model and design these properties, a distributed approach is often favoured in place of a centralized one. The choice of a distributed paradigm allows WSNs to act as a computational grid to perform complex processing that can not be managed by a single unit. In other words, the interaction among agents yields the emergence of collective behaviors, enacted via local rules, that globally regard the whole network and are beneficial in reaching the desired performance.

In the rest of the chapter, the interest is mainly focused on the so-called Industrial Wireless Sensor Networks (IWSNs) that have emerged as a key technology to improve the efficiency of production processes while limiting implementation costs in the competitive industrial marketplace. Recent developments have led to the realization of low-cost sensor nodes equipped with effective data processing and communication capabilities, thus enabling IWSNs to incorporate networks of up to thousands of these autonomous devices to facilitate the realization of highly reliable and self-healing intelligent systems for industrial applications (Güngör and Hancke (2009); Leitão, Mařík, and Vrba (2013); Hancke, Güngör, and Hancke (2014)).

The collaborative nature, rapid deployment and flexibility of IWSNs bring several advantages over traditional wired industrial systems. Specifically, the lack of cables, the support for mobility and the low-power maintenance make these solutions suitable for harsh environmental conditions (Åkerberg, Gidlund, and Björkman (2011)). The existing and potential applications of IWSNs can be classified according to the taxonomy in Güngör and Hancke (2013) into three main categories: *environmental sensing*, *condition* *monitoring*, and *process automation*. They regard a wide and heterogeneous range of specific scenarios that include building automation (Osterlind, Pramsten, Roberthson, Eriksson, Finne, and Voigt (2007)), process control (Chen, Cao, Cheng, Xiao, and Sun (2010)), utility automation (Fadel, Güngör, Nassef, Akkari, Malik, Almasri, and Akyildiz (2015)), precision agriculture (Ivanov, Bhargava, and Donnelly (2015)), to cite a few.

Besides real-time performance, IWSNs differ from other wireless sensor networks because of an increased attention towards maintainability, reliability, and safety (Güngör and Hancke (2009); Åkerberg et al. (2011)). In the industrial environment the networks should be able to guarantee robust operations in often critical scenarios and to ensure the safety of personnel, machinery and propriety as well as fast detection and recovery from malicious external attacks. Moreover, IWSNs should operate autonomously for such process and service monitoring. To this aim, fault detection algorithms must be developed that are able to identify sensors and actuators whose operating conditions are different from those expected (Neuzil, Kreibich, and Smid (2014)). In this venue, network partitioning strategies able to group nodes that exhibit similar behaviors can serve as a useful tool.

2.2 Clustering Task

In all the WSNs applicative contexts, there is typically an intelligent unit, be it centralized or distributed, that realizes the principal task through monitoring and control actions, thus requiring real-time information delivery over very large-scale systems. In this sense, locality in the communication can, on the one hand, ensure a rapid spreading of the information and, on the other, reduce the data interpretation complexity by filtering out the unnecessary information that regards non-neighboring nodes. Hence, grouping nodes into local clusters arises as a fundamental tool to enhance the network self-organization capabilities and improve the system autonomicity towards the fulfillment of collective goals. The development of effective clustering strategies specifically tailored for the industrial environment is in particular a key research area towards the realization of self-organizing, real-time, robust and secure wireless sensor networks to be deployed in smart industrial applications (Gholami and Brennan (2016); Ding, Tian, and Yu (2016); Zhang, Han, Feng, Cheng, Zhang, Tan, and Fu (2015); Chen, Yang, and McCann (2015)).

2.2.1 Literature Overview

The literature on clustering in sensor networks is quite vast and heterogeneous: it involves different concepts of *clustering* and covers several disciplines. It is therefore difficult to draw an even fairly complete state of the art: in this respect and with no aim of being exhaustive, a brief overview of clustering results is presented in the following.

As a general distinction, clustering problem in a wireless sensor network can be tackled by considering the topology (*network decomposition*) or the data gathered from the environment (*data clustering*). The first approach is generally considered in the design of communication algorithms and protocols, where the formation of clusters can lead to higher energy efficiency and reduced communication delays (Afsar and Tayarani-N (2014)). On the other hand, data-based network partitioning allows to reduce the computational load by taking into account similarities among nodes measurements in applications that deal with large amounts of information (Eyal, Keidar, and Rom (2011)).

Several partitioning algorithms have been proposed to accomplish the clustering task. In particular, the network decomposition problem can be tackled via well known procedures, which can be divided into heuristic, weighted, hierarchical and grid schemes. A comprehensive overview can be found in Abbasi and Younis (2007) and the references therein. Differently, in data clustering the focus is moved from the topology of the information sources to the information flow itself. Most techniques are based on the spatio-temporal correlation properties of the data gathered by the sensors (see e.g., Ma, Guo, Tian, and Ghanem (2011)). In all these works, one solution appears as particularly popular, in which nodes are partitioned into clusters and some of them, called *cluster-heads* (CHs), take the crucial role of aggregating the data coming from the other elements in the cluster and forwarding them to the base station.

The interest around clustering strategies for general purpose wireless sensor networks is well established; conversely, only recently the application of these techniques to IWSNs has been considered. The work of Gholami and Brennan (2016), for example, presents two clustering approaches to realize tracking of mobile nodes in IWSNs, a static one where clusters have a predefined number of elements and a dynamic one, which achieves the task more efficiently. The application of clustering strategies to solve environmental sensing and conditional monitoring tasks is the subject of the work of Ding et al. (2016), where a real-time clustering algorithm is applied to an industrial monitoring network to achieve risk assessment in an energy-efficient way. The detection of fault and anomalies in industrial sensing or process applications is another problem that can be tackled via clustering strategies. The work of Zhang et al. (2015) first adopts a well known network decomposition algorithm, namely the Low Energy Adaptive Clustering Hierarchy (LEACH) designed by Abbasi and Younis (2007), then implements a fault detection algorithm that exploits the presence of clusters. Chen et al. (2015), instead, develop an original clustering algorithm (Distributed Matching-based Grouping Algorithm, DMGA) to partition the network into strongly correlated groups of at least a predefined number of nodes; on this basis, then, a General Anomaly Detection (GAD) distributed procedure is developed that exploits data correlation for real-time recognition of anomaly conditions. Again, all these approaches rely on the presence of CHs to facilitate the clustering task.

2.2.2 Problem Formalization

This chapter deals with clustering strategies in IWSNs, however, differently from other approaches proposed in the literature, network decomposition and data clustering are here considered together, since both these aspects are crucial in ensuring the performance standards of industrial communications. In addition, the aim is to design a fully dis-

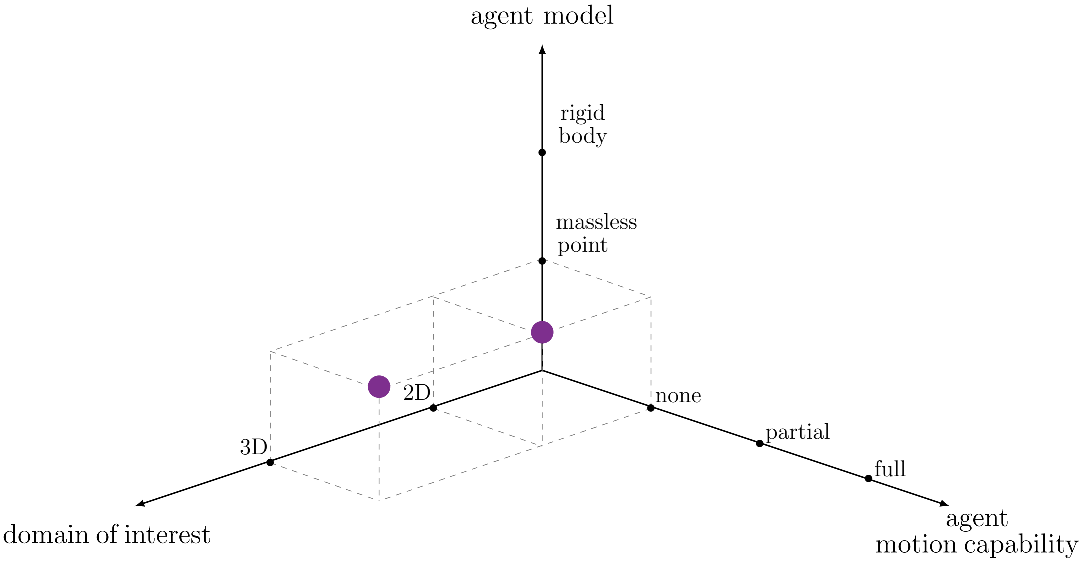


Figure 2.1. WSN: each wireless sensor node is modeled as a massless point having no motion capability and acting in the 2D or 3D space.

tributed approach that does not rely at all on the presence of CHs, yielding considerable benefits in terms of scalability and robustness.

Let consider an IWSN composed by n sensor nodes, whose topology is represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ is the set of bidirectional communication links among the agents (*communication graph*). Each element of the network is assumed to be able to sense a certain physical observable quantity, so that each node $v_i \in \mathcal{V}$ is associated with a scalar measurement $m_i \in \mathbb{R}$. In this sense, the domain of interest could be either a two-dimensional or a three-dimensional space: for instance, a set of sensor nodes can be exploited to monitor the pressure exerted on the surface of a press, as well as to control the temperature in a storehouse. In both the cases, the sensing devices have fixed position and can be modeled as a massless points as resumed in Figure 2.1.

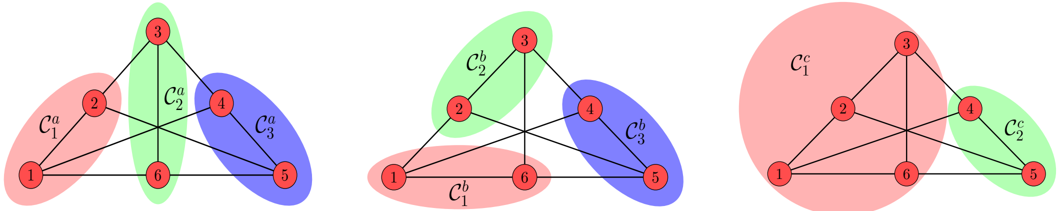
Stacking into vector $\mathbf{m} = [m_1 \; \dots \; m_n]^\top \in \mathbb{R}^n$ all the measurements gathered by the sensor nodes from the environment, the clustering task for a given WSN consists in identifying the node clusters $\{\mathcal{C}_\ell\}$ that constitute the unique partition of the node set $\mathcal{V}$, defined as $\mathscr{C} = \{\mathcal{C}_1 \dots \mathcal{C}_k\}$, satisfying the following criteria.

C1. **Connectivity:** $\forall\, \mathcal{C}_\ell \in \mathscr{C},\ \forall\, (v_i, v_j) \in \mathcal{C}_\ell \times \mathcal{C}_\ell\ \ \exists\ \ell_{ij} = \{v_i \dots v_j\}$.

C2. **Measurements similarity:** $\forall\, \mathcal{C}_\ell \in \mathscr{C},\ \forall\, v_i \in \mathcal{C}_\ell\ \exists\ v_j \in \mathcal{C}_\ell$ such that $|m_i - m_j| \leq b$; $b \in \mathbb{R}^+$ is named as *clustering bound.*

C3. **Maximality:** let $\mathscr{C}^i = \{\mathcal{C}^i_1, \dots, \mathcal{C}^i_{k_i}\},\ i \in \mathcal{I}$ be a generic partition of the network satisfying criteria C1-C2 ($\mathcal{I}$ is an arbitrary index set), the maximal partition is $\mathscr{C}^\star = \{\mathcal{C}_1 \dots \mathcal{C}_{k^\star}\}$, where $k^\star = \arg\min_{i \in \mathcal{I}}\ k_i$.

The elements of the obtained partition $\mathcal{C}_1 \dots \mathcal{C}_{k^\star}$ are denoted as *optimal clusters.* Let introduce the function $H \colon \mathcal{V} \to \mathscr{C}^\star$ that maps each node to the optimal cluster it belongs to in the optimal partition.



(a) Partition $\mathscr{C}^a$ with 3 clusters (C1-C2)

(b) Partition $\mathscr{C}^b$ with 3 clusters (C1-C2)

(c) Partition $\mathscr{C}^c$ with 2 clusters (C1-C2-C3)

Figure 2.2. Example of clustering on a 6-nodes network, where the measurement vector is $\mathbf{m} = [10 \;\; 12 \;\; 13 \;\; 20 \;\; 22 \;\; 11]^\top$ and the clustering bound is $b = 2$.

Remark 2.2.1. (Optimal) clusters are non-overlapping and cover the entire network, i.e., any node in $\mathcal{V}$ belongs to one and only one cluster following C1-C2-C3. This results in the function H to be bijective.

Looking at the three criteria, criterion C1 requires that each cluster forms a connected subgraph: in practice, in industrial applications, this means that the measurements and more in general the data obtained from nodes within the same cluster are shared. Criterion C2, instead, states that a sort of similarity exists among the measurements of nodes in a cluster: from the application point of view, the clustering bound b is a setup parameter related to the expected variance in the measurements range. Finally, since there can be several network partitions which fulfill C1 and C2, condition C3 is introduced to select the partition composed by the minimum number of clusters, i.e., the one wherein the cardinality of each cluster is maximal. This ensures that the defined partition is unique, apart from pathological cases.

Example 1. In order to provide an example of the presented clustering task and to show the uniqueness of the defined partition, a network composed by 6 nodes is considered in Figure 2.2. The following (scalar) measurement vector is assumed, $\mathbf{m} = [10 \;\; 12 \;\; 13 \;\; 20 \;\; 22 \;\; 11]^\top$, with a clustering bound chosen as $b = 2$, which corresponds to the admitted standard deviation for a *correct* measurement.

Figures 2.2a-2.2b show two partitions of the network, $\mathscr{C}^a$ and $\mathscr{C}^b$, both composed by three clusters, that satisfy criteria C1 and C2. Nevertheless, a further partition $\mathscr{C}^c$ with only two clusters can be identified, as reported in Figure 2.2c, which also abides by C1 and C2, and maximizes the cardinality of the clusters (C3).

Interestingly, with respect to C1 and C2 only, all these partitions show a cluster that includes exclusively nodes v_4 and v_5 due to the similarity between their measurements and the dissimilarity with the other nodes values; conversely, the remaining nodes show a higher level of measurement similarity among them and can be grouped according to different connectivity graphs that are identified in the network. From a building intelligence perspective, this suggests the possibility of detecting and isolating faulty nodes (in this case v_4 and v_5) or anomalous events through the clustering procedure.

♦

2.3 Clustering Algorithms

In this section two clustering algorithms are presented: the former is designed through a centralized approach, while the latter is achieved according to the distributed paradigm. Both strategies converge to the same solution providing an optimal network partition with respect to the established criteria.

2.3.1 Centralized Clustering Algorithm (CCA)

In the framework introduced in Section 2.2.2, it is possible to solve the clustering task through the centralized procedure reported in Algorithm 1. The inputs of the algorithm are the measurement vector $\mathbf{m}$, the clustering bound b and the (symmetric) adjacency matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ of the network. The outcome of the algorithm is instead the set $\{\mathcal{C}(v_i)\}_{i=1}^{n}$ which is related to the optimal partition $\mathscr{C}^\star$ defined in Section 2.2.2, through the neighborhood $\mathcal{N}_i$ of each node v_i, according to the relation $\mathcal{C}(v_i) = \mathcal{N}_i \cap H(v_i)$.

The proposed solution relies on the dynamic update of the (scalar) terms b_i^ℓ and b_i^u that indicate a lower and upper bound, respectively, associated to each node v_i. The initial part of the algorithm is devoted to the setup phase (rows 2-5): for each node v_i, $\mathcal{C}(v_i) = \{v_i\}$, while the two bounds b_i^ℓ and b_i^u are defined basing on the initial node measurement m_i and the imposed clustering bound b. The remaining part of Algorithm 1 consists of two steps that are iteratively repeated.

1. *Inclusion of nodes in clusters* (rows 8-16): for each node v_i, all the neighbors not already included in $\mathcal{C}(v_i)$, i.e., $v_j \in \mathcal{N}_i \backslash \mathcal{C}(v_i)$, are explored. Neighbor v_j is added to cluster $\mathcal{C}(v_i)$ only if the *measurements similarity criterion* (C2) is fulfilled.

2. *Update of bounds* (rows 17-32): in the second step, for each cluster $\mathcal{C}(v_i)$ whose cardinality is larger than 1, two (scalar) quantities are computed, namely the minimum lower bound b_{min}^ℓ and maximum upper bound b_{max}^u

   $$b_{min}^\ell = \min_{v_k \in \mathcal{C}(v_i)} b_k^\ell, \qquad b_{max}^u = \max_{v_k \in \mathcal{C}(v_i)} b_k^u.$$

   Then, every node in the cluster updates its bounds accordingly, i.e., $b_k^\ell = b_{min}^\ell$, $b_k^u = b_{max}^u \ \forall \ v_k \in \mathcal{C}(v_i)$.

The procedure ends when the second step does not produce any update of node bounds and, therefore, the cardinality of each cluster coincides with that of the previous iteration step.

Remark 2.3.1. Note that by adopting CCA, criteria C1, C2 and C3 are actually fulfilled. Indeed, a generic node v_j is inserted in $\mathcal{C}(v_i)$ through the instructions in rows 10-14. Specifically, this happens only if v_j is the neighbor of another node v_h in the same cluster, i.e., $v_j \in \mathcal{N}_h \backslash \mathcal{C}(v_i)$ and $v_h \in \mathcal{C}(v_i)$. At the same time, it is ensured that v_j is inserted in the cluster $\mathcal{C}(v_i)$ only if $\exists \ v_h \in \mathcal{C}(v_i)$, such that $b_h^\ell \leq m_j \leq b_h^u$, thus satisfying C2.

Algorithm 1 CCA

```
procedure CCA($\mathbf{m}$,$\mathbf{A}$,$b$)
    $term \leftarrow false$
    $\mathcal{C}(v_i) \leftarrow \{v_i\} \; \forall \; i$
    $b_i^\ell \leftarrow m_i - b \; \forall \; i$
    $b_i^u \leftarrow m_i + b \; \forall \; i$
    while not $term$ do
        $update \leftarrow false$
        for $i \leftarrow 1$ to $N$ do
            for $j \leftarrow 1$ to $N$ do
                if $v_j \in \mathcal{N}_i \backslash \mathcal{C}(v_i)$ then
                    if $\exists \; v_h \in \mathcal{C}(v_i) : b_h^\ell \leq m_j \leq b_h^u$ then
                        $\mathcal{C}(v_i) \leftarrow v_j$
                    end if
                end if
            end for
        end for
        for $i \leftarrow 1$ to $N$ do
            if $|\mathcal{C}(v_i)| > 1$ then
                $b_{min}^\ell \leftarrow \min_{v_k \in \mathcal{C}(v_i)} b_k^\ell$
                $b_{max}^u \leftarrow \max_{v_k \in \mathcal{C}(v_i)} b_k^u$
                for each $v_j \in \mathcal{C}(v_i)$ do
                    if $b_j^\ell > b_{min}^\ell$ then
                        $b_j^\ell \leftarrow b_{min}^\ell$
                        $update \leftarrow true$
                    end if
                    if $b_j^u < b_{max}^u$ then
                        $b_j^u \leftarrow b_{max}^u$
                        $update \leftarrow true$
                    end if
                end for
            end if
        end for
        $term \leftarrow$ not $update$
    end while
end procedure
```

Finally, Algorithm 1 stops only when there are no more bounds update ensuring that the corresponding partition has the lowest possible number of clusters, fulfilling C3.

It can be proved that the proposed procedure has complexity $O(n^3)$. Indeed, the instructions inside the *while* loop require $O(n^2)$ computations, due to the two nested *for* loops, whereas the *while* loop is executed n times in the worst case. This situation happens when the sensors form a line graph, i.e., $a_{ij} = [\mathbf{A}]_{ij} = 1 \Leftrightarrow |i - j| \leq 1$, and are provided with evenly spaced measurements at distance b, i.e., $m_i = (i-1)b$. In this scenario, the upper bound of node v_1 (which is the last one to converge together with v_n) at the k-th iteration is given by $b_1^u = (k+1)b$ and reaches the maximum value of nb only at iteration $k = n - 1$. The algorithm hence converges only at the subsequent iteration, thus yielding an overall $O(n^3)$ complexity. However, it should be noted that this worst-case scenario is quite uncommon to find in practice and therefore the complexity is generally lower.

2.3.2 Distributed Clustering Algorithm (DCA)

The centralized approach of CCA is based on the assumption that all nodes measurements are available at the same time together with the network topology at a single location. Although this statement might be true for some network configurations, it is not verified in a generic IWSN, and a distributed paradigm that relies only on local communication exchange is generally preferable. Moreover, the decentralized strategy results to be more robust to node failures and dynamic network topology modifications, which is a valuable feature in an industrial environment characterized by noise sources that may impair communication and faults due to the application to critical operational scenarios.

The proposed distributed algorithm (DCA) is reported in Algorithm 2 where the iterative nature of the procedure regards the execution of the same instructions by each node v_i of the network until the unique partition $\mathscr{C}^\star$ that fulfills criteria C1, C2 and C3 is determined. To this aim, a label c_i is associated to each node v_i to specify the cluster to which it belongs to (at the beginning $c_i = i \ \forall i$). This (scalar) variable is updated during the algorithm execution, so that the output of the whole procedure is a set of labels, one for each node, describing the partition of the network, in the sense that $c_i = c_j \Leftrightarrow H(v_i) = H(v_j)$. As in the centralized solution, a node v_i is also associated to a lower and an upper bound, b_i^ℓ and b_i^u respectively, that are initialized as in Algorithm 1 depending on the initial node measurement and the selected clustering bound.

DCA is again based on an iterative exploration of the neighbors, performed by each node in a distributed manner. In detail, node v_i checks the measurement of each node v_j belonging to its neighborhood $\mathcal{N}_i$ under the constraint $c_j \neq c_i$, meaning that they do not belong to the same cluster. If the two measurements are similar, according to criterion C2, then both the labels c_i and c_j are set to $\min(c_i, c_j)$ (rows 6-8). Moreover, node v_i keeps track of the minimum and maximum values assumed by b_k^ℓ and b_k^u respectively (rows 10-15) for any compatible neighbor v_k, in order to update its own bounds at the end of neighbors exploration (rows 19-23).

Subsequently, a further exploration of all the neighbors of v_i that belong to its same cluster, i.e. $c_j = c_i$, is performed to update their bounds accordingly (rows 24-33). The node v_i stops to perform the iterative execution of this algorithm when its flag *active* becomes true. This label is initialized to false at the beginning of the procedure and set to true in three possible cases: a new compatible neighbor is found, the node is included in another cluster or its bounds are updated.

The rationale behind DCA resides in the iterative bounds update and nodes inclusion into clusters, similarly to that of CCA (Algorithm 1); remarkably, though, DCA exploits only local information to attain the network partition and it is executed locally by each node, without requiring a central controller. Notably, for a given network and set of measurements, the two algorithms produce the same partition, the one defined in Section 2.2.2, as confirmed by extensive numerical simulations.

Example 2. Consider a synthetic IWSN composed of $n = 100$ nodes whose communication graph is described by a random geometric graph (introduced in Section 1.2.2),

Algorithm 2 DCA

```
procedure DCA($m_i$,$\mathcal{N}_i$,$b$)
    $active(v_i) \leftarrow false$
    $b^{\ell}_{min} \leftarrow \min_{v_h : c_h = c_i} b^{\ell}_h$
    $b^{u}_{max} \leftarrow \max_{v_h : c_h = c_i} b^{u}_h$
    for each $v_j \in \mathcal{N}_i,\ c_j \neq c_i$ do
        if $b^{\ell}_i \leq m_j \leq b^{u}_i$ then
            $c_j \leftarrow min\,(c_i, c_j)$
            $c_i \leftarrow c_j$
            if $b^{\ell}_{min} > b^{\ell}_j$ then
                $b^{\ell}_{min} \leftarrow b^{\ell}_j$
            end if
            if $b^{u}_{max} < b^{u}_j$ then
                $b^{u}_{max} \leftarrow b^{u}_j$
            end if
            $active(v_i) \leftarrow true$
            $active(v_j) \leftarrow true$
        end if
    end for
    if $b^{\ell}_{min} < b^{\ell}_i$ or $b^{u}_{max} > b^{u}_i$ then
        $b^{\ell}_i \leftarrow b^{\ell}_{min}$
        $b^{u}_i \leftarrow b^{u}_{max}$
        $active(v_i) \leftarrow true$
    end if
    for each $j \in \mathcal{N}_i,\ c_j = c_i$ do
        if $b^{\ell}_{min} < b^{\ell}_j$ then
            $b^{\ell}_j \leftarrow b^{\ell}_{min}$
            $active(v_j) \leftarrow true$
        end if
        if $b^{u}_{max} > b^{u}_j$ then
            $b^{u}_j \leftarrow b^{u}_{max}$
            $active(v_j) \leftarrow true$
        end if
    end for
end procedure
```

whereas measurements are randomly drawn from a uniform distribution with range between 0 and 100, $m_i \in \mathcal{U}\left([0, 100]\right)$, and the clustering bound is set to $b = 5$. One realization of such scenario is reported in Figure 2.3 where both the network topology (left plot) and the convergence behavior of DCA in terms of number of clusters (right plot) are shown.

The initialization of DCA corresponds to the creation of one cluster per node and the iterative procedure makes the number of clusters decrease as nodes are merged together, according to the update of the node bounds. In this context, a step is defined as the full execution of Algorithm 2 by every node in the network. As expected, the result of the distributed procedure converges to the centralized solution after three steps. Specifically, in the reported case, the network optimal partition $\mathscr{C}^{\star}$ is constituted by 2 clusters.

♦

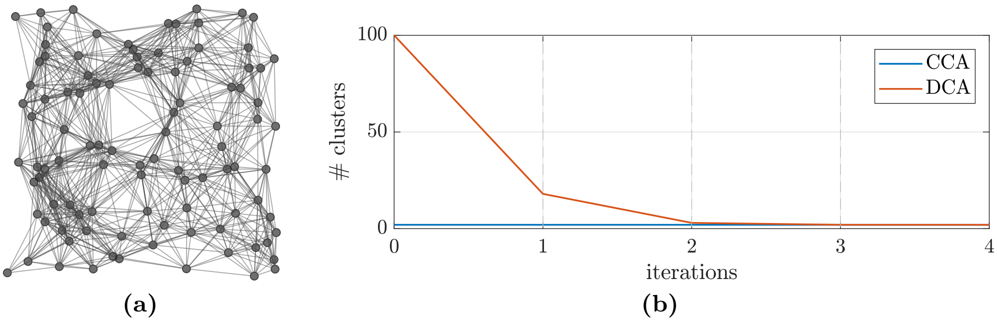


Figure 2.3. Example of clustering on a 100-nodes network represented by a random geometric graph (a). The DCA solution converge to the one yielded by CCA procedure (b).

2.3.3 First Assessment of DCA

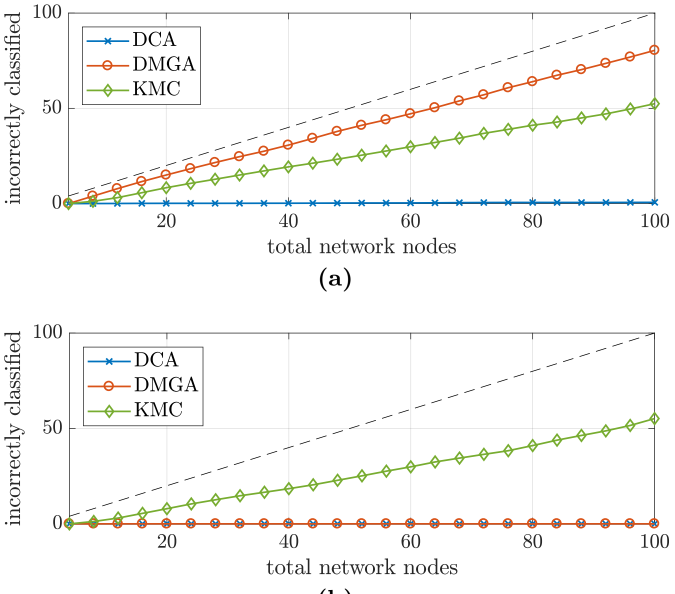
Here, the performance of the distributed partitioning strategy DCA is compared with two other procedures: the k-means clustering approach (hereafter, KMC) and the DMGA strategy. The first one is a well-known partitioning algorithm that aims at grouping the nodes of a network into a fixed number k of clusters evaluating the distance from a priori set centroids, while the second one is the clustering solution described in Section IV-A of Chen et al. (2015) which relies on the maximization of the spatio-temporal correlation between pair of sensors.

The evaluation is conducted on a network made of n nodes with n ranging from 8 to 100, organized into n_C clusters of fixed size equal to 4 (hence, n_C spans from 2 to 25). The measurement of a generic node in the i-th cluster is uniformly distributed in the range $[\bar{m}_i - b/2, \bar{m}_i + b/2]$, where $\bar{m}_i$ is the cluster average measurement and it is generated as $\bar{m}_i = \bar{m}_0 + 2b(j-1)$ with $j = 1 \dots n_C$ (b is the clustering bound). Two different network topologies are considered: in one case, a random geometric graph is chosen within each cluster and any two clusters are connected by one link between two CH nodes with probability $p_L = 0.9$ (Figure 2.4a); in the other case, instead, the communication network is described by a complete graph (Figure 2.4b). In both cases, the number of misclassified nodes is considered versus the network size. Indicating with $\mathscr{C}^\star$ the optimal partition yielded by CCA and with $\mathscr{C}$ a generic partition, the misclassification cost function is given by

$$d\left(\mathscr{C}, \mathscr{C}^\star\right) = \sum_{i=1}^{n} I_i,$$

where I_i is an indicator function that is equal to zero if the set of cluster elements of node v_i in partitions $\mathscr{C}$ and $\mathscr{C}^\star$ coincides (node v_i is correctly classified) and it is equal to one otherwise (node v_i is misclassified).

For KMC, the parameter k is set equal to n_C and the centroids are randomly initialized. For DMGA, the correlation coefficient ϱ_{ij} to maximize is defined as the ratio between the distance of two measurements and the measurements range, i.e., $\varrho_{ij} = 1 - \frac{|m_i - m_j|}{(2n_C - 1)b}$. In addition, the minimal correlation group size is chosen equal to $n_{min} = n_C = 4$ and the minimal correlation threshold is imposed to $\varrho_{min} = 1 - (2n_C - 1)^{-1}$.



(a)

(b)

Figure 2.4. Node classification comparison: (a) non-fully connected network topology; (b) fully connected network topology.

Figure 2.4 shows how the performance of KMC are totally independent of the network topology as such strategy is exclusively based on the similarity of the measurements and does not take into account the communication links. Specifically, the fact that the algorithm neglects the structure of the network and the consequent totally random initialization of centroids cause KMC to get stuck on local optima, which results in the significant percentage of misclassified nodes observable in Figure 2.4. On the contrary, applying the DMGA procedure, the set of nodes that are incorrectly classified is empty when the network is modeled through a complete graph, while a significant number of nodes is not classified appropriately when a less connected communication structure is adopted. Instead, and most remarkably, the partition provided by DCA algorithm coincides with the optimal one in both cases, thanks to the fact that the proposed procedure considers at the same time both the IWSN connectivity properties and the measurements similarity. Moreover, as explained in Sections 2.3.1 and 2.3.2, CCA and DCA always yield the same clustering solution, which is equal to the optimal one defined in Section 2.2.2.

2.4 Application to the Industrial Scenario

Many practical applications of the clustering procedure can be envisaged within the context of IWSNs. They range from the fault detection along a production line to the monitoring of a structured environment in building or factory automation, from the tracking of mobile nodes in a productive industrial area to the optimization of energy resources for a surveillance system. In this sense, two different real-world scenarios have been considered for validation of the proposed DCA, specifically:

#1 a factory process line, where an item undergoes several production stages on its way from raw material to end product;

#2 a structured indoor environment, wherein the task is that of indoor monitoring, since building energy management issues may arise and anomaly detection is also important.

Scenario #1 concerns the conditions monitoring of highly dynamic processes in modern factory facilities, where the timing behaviors of the control variables and of the quantities of interest need to be accurately monitored in order to ensure efficiency, performance and quality to the process/service. A factory intelligence unit should manage to follow the product/process chain, to identify the different stages, and to detect possible faults and anomalies that may occur. Indeed, this issue can be experienced in a large variety of production plants and processes. Just to provide a couple of examples: in the context of the food industry, the traceability of the product and the proper management of the ambient conditions throughout the whole supply chain are of paramount importance to guarantee the quality and safety of products and to extend their shelf life (Piramuthu and Zhou (2016)). In semiconductor manufacturing, the development of intelligent monitoring systems based on IWSNs can increase the automation and maintainability of such complex process, thus leading to real-time problem diagnostics and production optimization procedures (Susto, Pampuri, Schirru, Beghi, and De Nicolao (2015)).

Scenario #2 is characteristic of many industrial, commercial, and public service installations, and refers to environmental sensing and service monitoring. In particular, service monitoring aims at offering to the end–users a designed (or expected) quality of service, and proposing to the providers a tool to control and optimize the use of resources and increment the awareness of their employment. These instances are strictly related to building automation, which has received a surge of attention in the last few years towards the deployment of green building solutions in the industry (Güngör and Hancke (2013); Fadel et al. (2015)). Environmental sensing, instead, is referred to the task of measuring quantities that can be only partially controlled but are fundamental for the efficiency of equipment and operators, to detect pollution, avoid hazard and ensure security, and also yield comfort in the workspace (Ma et al. (2011)).

In detail, the focus in this section is posed on the validation of the distributed strategy DCA, since it has already been shown to converge to the centralized solution, and it represents the most interesting approach for practical applications due to its intrinsic robustness and flexibility. Thus, the performance of the proposed partitioning procedure is studied with respect to different kind of faults in the first scenario, while it is compared with those of KMC and of DMGA/GAD algorithms in the second case.

2.4.1 Numerical Validation: Process Monitoring Application

This scenario is schematically represented in Figure 2.5, where a plant with multiple production stages is considered, and a number of items move along the production line.

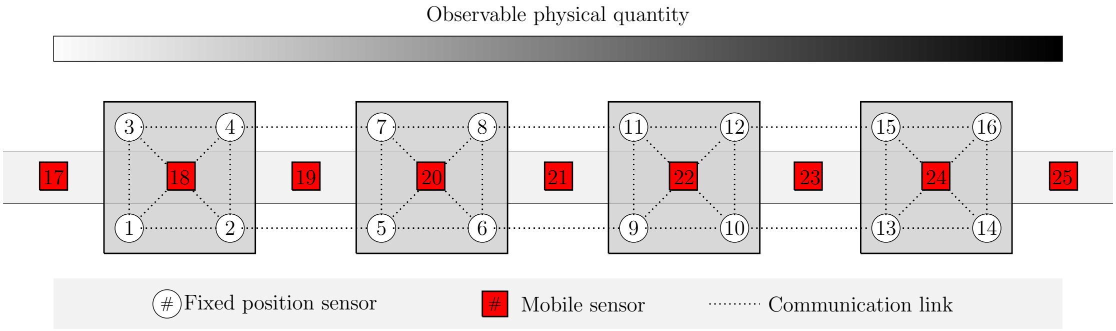


Figure 2.5. Application scenario #1 - industrial line-production process characterized by $s = 4$ stages (gray boxes) each monitored by $n_s = 4$ sensors (white numbered circles), and several moving items (depicted as red squares), which are associated to measurement sensors.

Each stage is assumed to be characterized by a specific value of an observable physical quantity (e.g., temperature, pressure, vibration, or a combination of heterogeneous variables): in this respect, a gradient of such quantity can be measured across the different production phases and a monitoring IWSN is installed, made up by different subnetworks, each corresponding to a distinct production phase. Also, the item that is undergoing the production process is equipped with or accompanied by a wireless sensor node, so that its own local measurement can be taken along the production line.

In correctly working conditions, the fixed sensor nodes would be grouped into four clusters, corresponding to the different production stages, and the mobile node associated to the item can be inserted into a specific cluster, meaning that it is undergoing the corresponding stage, or grouped by itself, when traveling from one stage to another. It follows that, since the expected outcome of the clustering procedure for a mobile node is a priori known, any deviation from such behavior can be seen as an anomaly and detected by an intelligent supervisory system, which may be distributed as well.

In this context, DCA is able to keep track of the item state and detect possible anomalies, given the fact that it takes into account both communication network topology and measurements distribution. This detection can be actually based on a simple *cluster label* comparison, by opportunely choosing the IDs for the fixed and the mobile nodes. With reference to the example in Figure 2.5, all fixed sensors have a lower ID with respect to the mobile nodes: since any node can retrieve the cluster it is included in by looking at its the *cluster label* (that assumes the value of the lowest node ID in the same cluster), a mobile node can recognize if it is grouped alone (cluster label equal to its ID) or grouped in a specific cluster (cluster label lower than its ID). Consequently, since the evolution of its cluster label over time can be a priori stored in the node memory, the node itself can autonomously detect an anomaly when the actual evolution differs from the expectation.

The application of DCA to this framework allows to identify several different anomalies. In particular, in this context the following ones are considered.

- **Measurement fault:** the value of the observed quantity measured by the mobile node at one stage is significantly different from that expected. In this case, the node clusters by itself when it should be clustered with the stage nodes.

- **Timing mismatch:** the mobile node reaches a certain stage earlier or later than expected. In this case, the evolution of the cluster label is anticipated or delayed with respect to the nominal trend.

- **Communication fault:** the mobile node experiences communication loss and is no longer able to exchange messages with other nodes. In this case, the node is always grouped by itself.

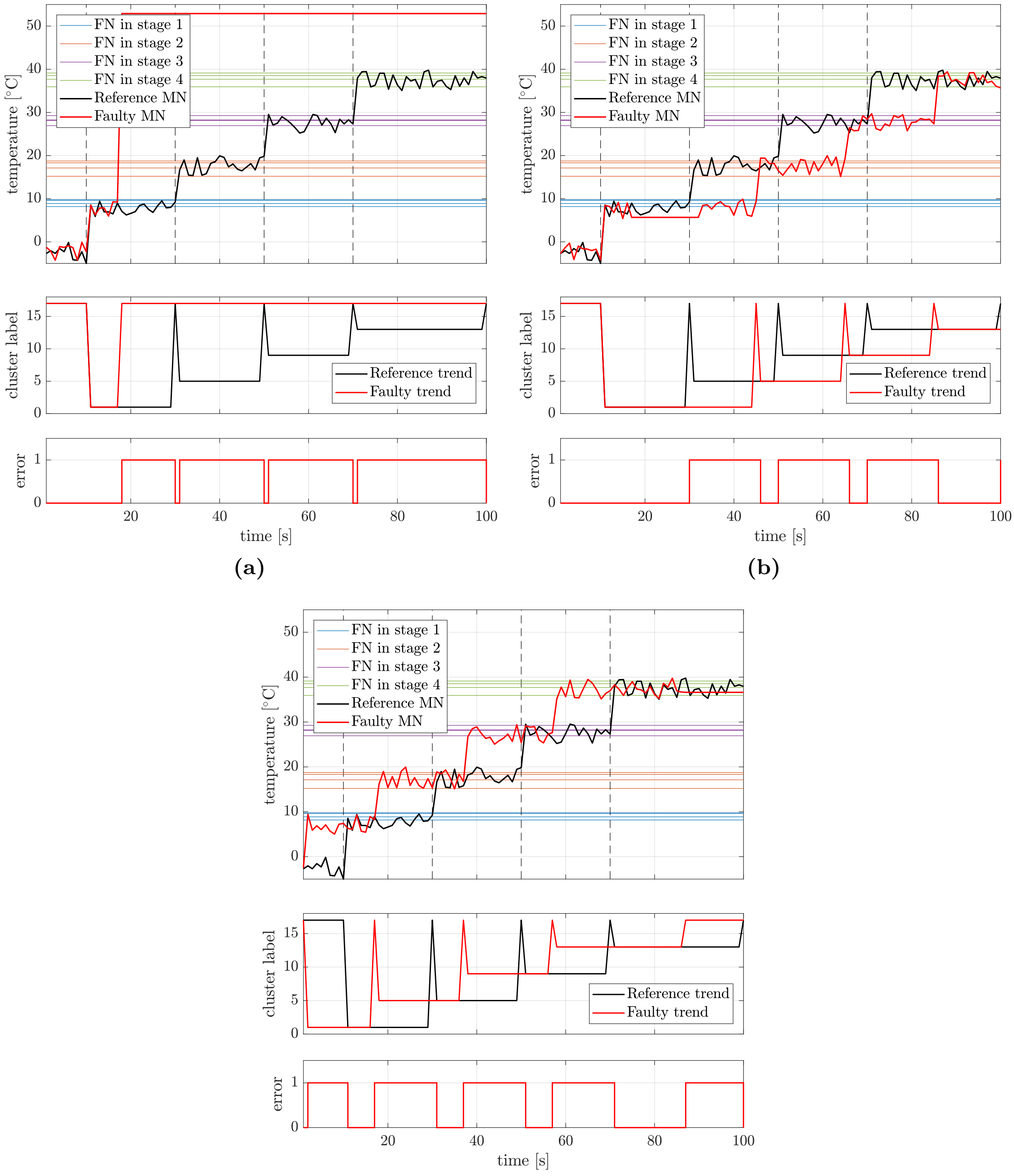
The intelligent supervisory system behavior in presence of these types of anomalies has been simulated for different scenarios, with multiple fault instances and increasing network complexity ranging from tens to thousands of sensor nodes: the results of these numerical experiments consistently show that the proposed method always allows to detect the different kinds of anomalies with no occurrence of false positives.

An example in this sense is reported in Figure 2.6 for a mobile node that experiences a failure, in the scenario represented in Figure 2.5: here, the observed physical quantity is the process temperature and $s = 4$ production stages are characterized by temperature ranges $\Delta T_1 = [5, 10]^\circ$ C, $\Delta T_2 = [15, 20]^\circ$ C, $\Delta T_3 = [25, 30]^\circ$ C and $\Delta T_4 = [35, 40]^\circ$ C, while the initial temperature measured by the mobile node is below 0 °C (temperature of the raw material before the process). A value of $b = 4\,^\circ$C is selected as the clustering bound. The resulting system behaviors (actually, referring to one realization of such scenario) are shown in terms of measurements trends, clusters evolution, and fault detection signal. This latter signal is obtained as the mismatch (computed in practice via logical XOR) between the actual clusters evolution value with the reference expected one. Interestingly, in the case of communication fault, namely loss of signal from the monitoring mobile station, the measurement evolution of the mobile node is statistically close to the reference one, to the point that no anomaly can be identified by looking only at the measurements, while the cluster label comparison promptly reveals the anomaly.

2.4.2 Experimental Dataset: Environmental Sensing Application

The DCA approach has also been applied to a dataset coming from a sparse sensor network deployed in a public indoor environment characterized by heterogeneous usage zones, with reference to Figure 2.7, respectively Area 1 to 4. The considered monitoring network is composed by $n = 17$ wireless T-mote nodes (Moteiv Corporation (2006)) allocated in 4 different areas composed of multiple rooms and a sample connection is assumed as shown in the graph reported in Figure 2.7a. This dataset has been derived from a 4 months monitoring period that includes weekends and holidays; in detail, each sensor measures a temperature with a fixed sampling interval of five minutes.

In this context, the application of DCA to a set of static measurements collected at a specific time instant (e.g., at 12 a.m. of a generic weekday) with a suitable clustering bound b yields a network partition such that there is a two–way correspondence between clusters and areas, as reported in Figure 2.7a. This cluster-area correspondence is not achieved by partitioning the network through KMC and DMGA strategies (see Figure 2.7b).



(a)

(b)

(c)

Figure 2.6. Application scenario #1 - system behavior in presence of anomalies: (a) measurement fault, (b) timing mismatch and (c) communication error. For each anomaly, it is reported the measurements of fixed nodes (FN) and mobile node (MN) in the top row, the clustering trends in the middle row, and the fault detection signal in the bottom row.

Indeed, even if with $k = 4$ KMC identifies four clusters, they do not coincide with the different areas of the building, mainly because of the reduced signal variability in the whole environment. On the other hand, the implementation of DMGA with $n_{min} = 2$ to allow for the identification of small groups, leads to seven clusters because of the network sparsity. Remarkably, DCA strategy can handle both of these aspects correctly detecting the four areas, which are characterized by a specific measurement behavior and, hence,

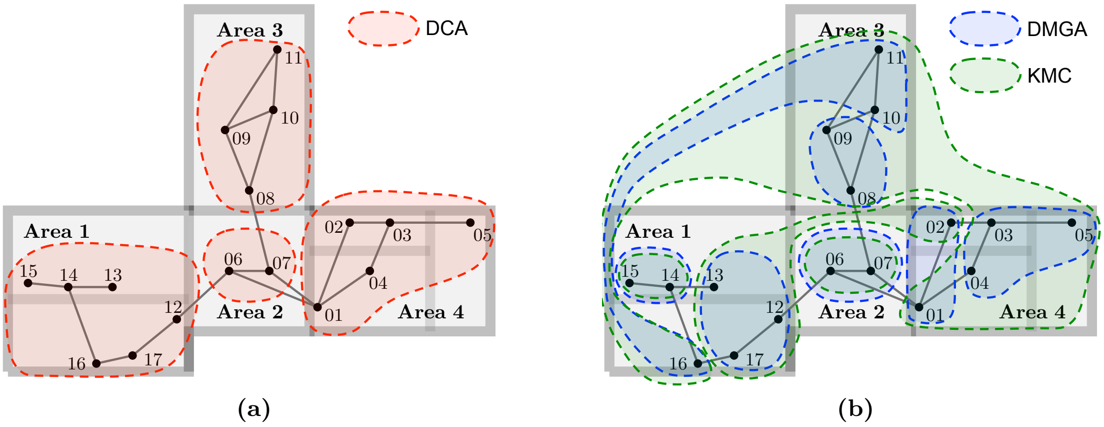


Figure 2.7. Application scenario #2 - (a) DCA network partition, (b) KMC (green) and DMGA (blue) network partitions.

may be managed in a dedicated way by an intelligent environment controller.

Static data processing, however, may be not informative enough for a building management system with the aim of energy profile optimization and efficiency. In this sense, the application of the clustering algorithm during a large observation interval allows to extract general trends and to build an effective anomaly detection strategy. In order to validate this claim, anomalous values are artificially inserted in the measurements collected by the network during 8 weeks of operations. In particular, two anomaly models are here considered (Sharma, Golubchik, and Govindan (2010)):

- **constant**: the sensor reports a constant value that corresponds to the measurement at the beginning of the fault period multiplied by a factor $\gamma = 2$;

- **noise** : the sensor measurement is affected by an additive Gaussian noise with zero mean and standard deviation $\sigma = 10°\mathrm{C}$.

Different anomaly occurrence rates have been considered and the duration of each fault is a uniform random variable with an average value of 12 samples (1 hour).

In this framework, a simple yet effective fault detection strategy based on DCA is adopted: the algorithm runs in a supervised manner during the first week of operation, without any fault, and, at each sample, every node stores the list of its neighbors, i.e., the nodes in its same cluster. Then, in the following weeks, DCA is applied at any new measurements (with a possible presence of faults) and each node compares the detected neighbors with those stored for the corresponding instant of the training week: if less than 50% of current neighbors are not in the list, the node self-declares as faulty.

To evaluate the performance of this strategy against a state-of-the-art anomaly detection scheme, GAD algorithm (Chen et al. (2015)) is applied to the same dataset. The anomaly detection phase starts after the first week and the following parameters are used: $n_{min} = 2$ (minimal correlation group size), $\varrho_{min} = 0.5$ (minimal correlation threshold) and $\delta_t = 90$ samples (7.5 hours). Note that the choice of setting the minimum number of cluster elements to 2 is imposed by the considered scenario, characterized by relatively few

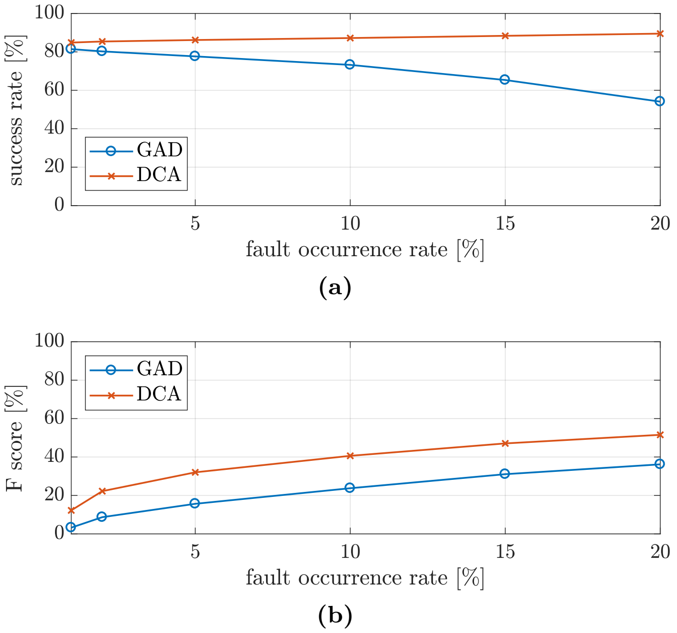


Figure 2.8. Application scenario #2 - (a) anomaly detection accuracy; (b) F-score.

nodes, sparsely connected and with a significant measurement variability. However, this choice strongly affects the detection capabilities of GAD, which generally requires denser clusters. To cope with this issue, the algorithm has been slightly modified, labeling as faulty all the nodes whose status is uncertain. This conservative choice is motivated by the significant robustness required by industrial sensing applications, which translates into the urge to detect as many faults as possible, enduring a possible high occurrence of false positives. Following a similar reasoning, the bound of the DCA algorithm is set to an adequately low value ($b = 2°\text{C}$), so as to privilege detection of faults with respect to false positives.

The performance comparison between the two anomaly detection strategies is shown in Figure 2.8 for several values of occurrence rate. Figure 2.8a reports the percentage of anomalies detected on the total number of generated ones. The solution based on DCA outperforms GAD for every value of anomaly rate, yielding an accuracy always greater than 85%, while using GAD it drops below 60% for a fault rate of 20%. To highlight how both strategies suffer from the presence of false positives due to the significant irregularity of the measurement trends, the F-score, a metric that sums up accuracy and precision (Van Rijsbergen (2004)), is depicted in Figure 2.8b. It can be observed that this metric is low, especially for low anomaly occurrence rates, where the incidence of false positives is significant. However, also from this point of view, the anomaly detection strategy based on DCA confirms its validity, outperforming GAD by a wide margin.

2.5 Results Summary

Within the framework of distributed intelligent systems, this chapter deals with wireless sensor networks that arise as a strategic multi-agent technology to be exploited in the

	DCA	**KMC**	**DGMA**
parameters to initialize	b (clustering bound)	k (number of clusters) centroids position	n_{min} (min. corr. group size) ϱ_{min} (min. corr. threshold)
topology dependence	yes	no	yes
measurements dependence	yes	yes	yes

Table 2.1. A table recalling the main features of the clustering algorithms considered in this chapter.

industrial automated context in order to realize smarter factories. In particular, the attention is here focused on the design of strategies to effectively partition a given network into non-overlapping clusters of nodes. To this purpose, three clustering criteria are proposed, that take into account both communication network topology and the measurements gathered by the sensor nodes. Indeed, these features results to be important in noisy industrial environments, where both the network connectivity and the measurement consistency concur to guarantee IWSN performance in terms of timeliness, reliability and security. In order to accomplish this task, two clustering strategies are proposed following either a centralized or a distributed approach, the former (CCA) relying on the presence of a central coordinating unit, the latter (DCA) employing the network itself as a computational grid, without the need of identifying CHs.

Effectively, the distributed solution converges to the centralized one after some iterations and hence emerges as the preferred one for IWSNs thanks to the inherent properties of autonomicity, scalability and robustness. The proposed DCA procedure is then tested both in numerical simulations and on a real-world dataset, to provide an assessment of its performance in environmental monitoring and fault detection applications employed in building and process automation. In this evaluation, the performance of DCA are compared with both a classical KMC approach and the most recent DGMA procedure (the main features of these strategies are listed in Table 2.1), leading to conclude that the proposed algorithm outperforms other approaches in accomplishing the clustering task and can be used, for example, as a basis to develop efficient anomaly and fault detection strategies to be employed in intelligent industrial monitoring applications.

3

Attitude Estimation for VSNs

Recently, the development of low-cost high-resolution imaging sensors and the widespread of IoT technologies are leading to the development of new applications that transform the traditional vision systems into ubiquitous and pervading visual sensor networks. Such systems integrate a large number of cameras that interoperate to perform some high-level tasks, such as the monitoring and interpretation of the environment or the control of actuators operating in it. The effectiveness of such tasks is significantly affected by the precision of the camera localization with respect to an inertial reference system, namely on the accuracy in the reconstruction of its pose (position and orientation).

In this context, this chapter tackles the distributed attitude estimation problem for a camera network in a realistic noisy environment. This stands as interesting issue for both the mathematical formalization as an optimization problem over a Riemannian manifold and the algorithmic solutions that can be derived. In addition, it constitutes an essential step in the development of an autonomous surveillance system.

3.1 Visual Sensor Networks

A Visual Sensor Network (VSN) is a multi-agent system constituted of a collection of spatially distributed smart camera-devices capable of processing and fusing images of a scene from a variety of viewpoints into some form more useful than individual images (Rinner and Wolf (2008); Bhanu, Ravishankar, Roy-Chowdhury, Aghajan, and Terzopoulos (2011); Kyung (2015)). Each camera, termed *smart* (or *intelligent*), consists of a vision system which, in addition to image capture circuitry, is capable of extracting application-specific information from the captured frames and sharing them though a communication interface within a context of intelligent and automated vision system (Belbachir (2010)).

One of the main differences between visual sensor networks and other types of sensor networks lies in the nature and amount of information that the agents perceive from the environment. Most sensors (including wireless sensors considered in Chapter 2) provide measurements as one-dimensional data signals, whereas the visual sensors usually cope with two-dimensional images. The additional dimensionality of the dataset results in richer information content as well as in a higher complexity of data processing, analysis and communication (Soro and Heinzelman (2009)). Nevertheless, thanks to the advances in high-performance embedded microcontroller, optimized computer vision techniques and reliable communication protocols, multi-camera networks have increasingly spread out during the last twenty years becoming ubiquitous smart systems in industrial, civil and domestic context. By acquiring information-rich data, these architectures enable vision-based interpretative applications as the intelligent IoT surveillance (Bramberger, Doblander, Maier, Rinner, and Schwabach (2006); Pavithra and Kathirvel (2017)), the domotics for smart living environments (Gruenwedel, Jelaca, Niño-Castañeda, Van Hese, Van Cauwelaert, Veelaert, and Philips (2012); Jalal, Uddin, and Kim (2012)), the assisted driving of autonomous vehicles (Bonin-Font, Ortiz, and Oliver (2008); Janai, Güney, Behl, and Geiger (2017)), the activity monitoring and control in industrial environments (Bleser, Pastarmov, and Stricker (2005); Di Leo, Liguori, Pietrosanto, and Sommella (2017)), to cite a few.

In all the mentioned applicative scenarios, it is mandatory the real-time process of the captured data in order to allow a timely action in case of a negative event (e.g., the malfunctioning of a machinery, the presence of an intruder), hence some level of automation in the whole camera system would be highly desirable and beneficial. In addition, in a time-saving optic, it is advisable the adoption of a distributed paradigm according to which the task accomplishment rests upon the interaction and the information exchange among the agents of the group. In this sense, although their substantial versatility, the visual multi-agent technologies are affected by several critical issues on the hardware, software, and system design aspect. On the hardware side, camera networks require energy and computationally efficient nodes able to capture and process visual information in a narrow time interval (Seema and Reisslein (2011)). On the software side, scalable, robust, and computationally efficient video/image analysis methods have to be implemented on these systems which are required to be able to share information among themselves (Micheloni, Rinner, and Foresti (2010); Regazzoni, Cavallaro, Wu, Konrad, and Hampapur (2010)). Finally, adaptive control and coordination algorithms are needed in smart VSNs to increase flexibility, ease deployment, and manage the middleware that links different sensor control modules (Roy-Chowdhury and Song (2012); Mohamed and Al-Jaroodi (2011)). A general body of knowledge around many of these issues exists, nevertheless multi-camera networks still constitute a fascinating area for current research.

3.2 Attitude Estimation Task

The efficiency of most of the tasks that a smart camera system can perform (as, for instance, perimeter surveillance, urban traffic monitoring) is strictly conditioned by the knowledge from each device of its location with respect to a global inertial reference system. Consequently, the self-estimation of the position and the attitude of each visual node composing a multi-device network constitutes a classic problem of the literature devoted to the autonomous multi-camera systems (see, e.g., Hemayed (2003); Bajramovic (2010) and the references therein).

This issue, that represents a critical first step in any multi-agent network deployment (Piovan, Shames, Fidan, Bullo, and Anderson (2013)), is typically solved starting from noisy relative measurements and recasting it as an optimization problem constrained to the topological structure of the network. The most challenging aspect in the determination of the pose of each device in the network is constituted by the recovery of each camera orientation with respect to a common frame, namely the attitude estimation problem, due to the nature of the considered variables which belong to a Riemannian manifold characterized by a proper mathematics.

3.2.1 Literature Overview

For a camera-device, the attitude estimation coincides with the well-known *localization* problem restricted to only the orientation component of its pose. Localization, indeed, involves also the estimation of the position in the inertial frame. In turn, the localization task is one step of the more complex *calibration* issue that entails the estimation of both the extrinsic parameters (i.e., position and attitude in the inertial frame) and intrinsic parameters (i.e., focal length, image sensor format and lens distortion) for the device. Nevertheless, in the following, the terms 'calibration' and 'localization' are equivalently used to indicate the camera (full) pose estimation problem, as often occurs in literature.

The calibration problem for a camera network, which trivially deals with the localization of each device in the system, has been broadly addressed in the past. Traditional techniques rely on the correspondence between the observed 3D points and their projection on the image plane of the visual devices: the extrinsic (and intrinsic) parameters of the each camera are derived by minimizing the re-projection errors, i.e., the image distances between the projected and measured points.

In this framework, a typical approach envisages the use of a target object (such as a checkerboard pattern), whose physical parameters and location in the inertial frame are known, that is simultaneously observed by all nodes of the camera system (Zhao and Liu (2008); Courchay, Dalalyan, Keriven, and Sturm (2010); Shen and Hornsey (2011)). In addition, to cope with situations of minimal or no overlap in the camera visual planes, the target object is usually observed exploiting planar mirror reflections (Kumar, Ilie, Frahm, and Pollefeys (2008); Rodrigues, Barreto, and Nunes (2010)).

These procedures require the human intervention, and are meant to be carried as an initial off-line step before starting operating the system. Motivated by the fact that manual ad-hoc localization methods are not suitable to handle large number of cameras or dynamic configurations in the network, the automation of the pose reconstruction process has become essential to cope with accuracy and real-time performance, so that the related literature is plentiful and still increasing. Specifically, various approaches have been developed according to the dimension of the estimation domain (2D vs 3D), the adopted solution paradigm (centralized vs distributed), the available measurements (e.g. distances, bearings, angles of arrival) and some a priori information (for instance, the a priori knowledge of the pose of some special node, referred as *beacons* or *anchors*). To provide a complete overview about the existing self-calibration techniques for a camera network is outside the scopes of this dissertation, however some interesting works are mentioned in the following in order to contextualize the proposed solution with respect to the current state of the art.

In the smart multi-agent networked context, many distributed algorithms still exploit the images correspondence combined with topological information. For example, Mantzel, Choi, and Baraniuk (2004) have proposed an iterative procedure which alternates between the triangulation of 2D image projections into the 3D space and the neighboring camera matrices estimation based on the assumed image-to-world correspondences. Similarly, Devarajan, Radke, and Chung (2006) have designed a distributed calibration strategy based on the exchange of image projection of common 3D points among cameras having overlapping visual fields following an incremental clustering approach. An incremental technique is also proposed in Aghajan and Cavallaro (2009), where triplets of cameras are considered each time.

Another very common approach to face the self-calibration of a camera system rests upon the analysis of the dynamics of the observed scene. When a dynamic framework is considered, a camera network can indeed be automatically calibrated by tracking a moving object as proved in Rahimi, Dunagan, and Darrell (2004); Funiak, Guestrin, Paskin, and Sukthankar (2006); Pflugfelder and Bischof (2010). The idea is to address the system calibration by solving a simultaneous localization and tracking problem, estimating then both the trajectory of the observed object and the poses of the devices.

From a more mathematical perspective, it can be observed that the automated calibration task can be recast as an optimization (or a consensus) problem over a Riemannian manifold (see Sarlette and Sepulchre (2009), Absil, Mahony, and Sepulchre (2009), Tron, Afsari, and Vidal (2013)). Conversely to standard calibration techniques, in this case,

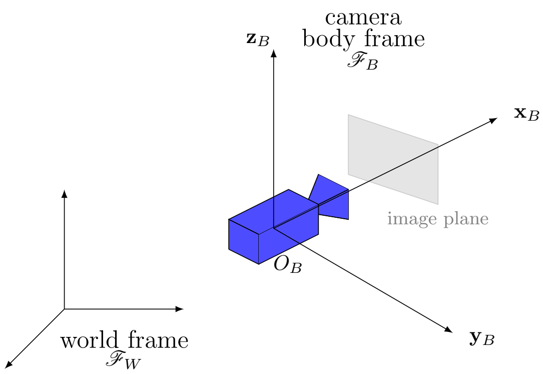


Figure 3.1. Simplification of the pinhole camera model: $\mathbf{x}_B$-axis identifies the direction of the camera field of view.

the optimization is carried out in the natural space of the problem, i.e., the space of rigid-body transformations. Some interesting results in this direction are presented in Tron and Vidal (2009, 2014), whose approach consists in an iterative procedure based on the minimization of a suitable cost function through a distributed strategy working in the Riemannian consensus framework; in Knuth and Barooah (2013b), which extends and refines the previous method by exploiting maximum-likelihood estimation techniques; in Knuth and Barooah (2012, 2013a), which deal with a dynamic scenario consisting of mobile agents; in Borra, Lovisari, Carli, Fagnani, and Zampieri (2012); Piovan et al. (2013), which narrow the problem to the planar (2D) case.

3.2.2 Problem Formalization

This chapter focuses on the orientation self-estimation problem for an homogeneous VSN. The determination of the attitude of each device in the system with respect to a certain global inertial system is addressed exploiting some noisy available measurements derived by the observed scene and performing the minimization of a suitable cost function through a Riemannian consensus strategy.

To do so, let introduce the global inertial frame $\mathscr{F}_W$. The attitude of a given camera in $\mathscr{F}_W$ depends on the definition of the body frame $\mathscr{F}_B = \{O_B, (\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B)\}$ in-built with the device. In particular, according to the most diffused pinhole camera model (Ma, Soatto, Kosecka, and Sastry (2004)), $\mathscr{F}_B$ is so that the origin O_B coincides with the center of projection of the camera, while the $\mathbf{x}_B$-axis (named optical or principal axis) is pointed in the viewing direction, the $\mathbf{z}_B$-axis is up faced and the $\mathbf{y}_B$-axis is oriented according to the left-handed coordinate system (see Figure 3.1). The attitude of a camera is thus identified by the rotation matrix that describes the orientation of the camera frame $\mathscr{F}_B$ with respect to $\mathscr{F}_W$. More formally, it is given by the matrix $\mathbf{R} \in SO(3)$ belonging to the three-dimensional Special Orthogonal group $SO(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}_3,\ \det(\mathbf{R}) = +1\}$ and such that $\mathbf{v}^W = \mathbf{R}\mathbf{v}^B$ being $\mathbf{v}^W \in \mathbb{R}^3$ a vector in the world frame and $\mathbf{v}^B \in \mathbb{R}^3$ the same vector in the camera frame[1].

[1] This convention corresponds to the ‘reference interpretation’ introduced in Section 5.3.4 of Tron (2012).

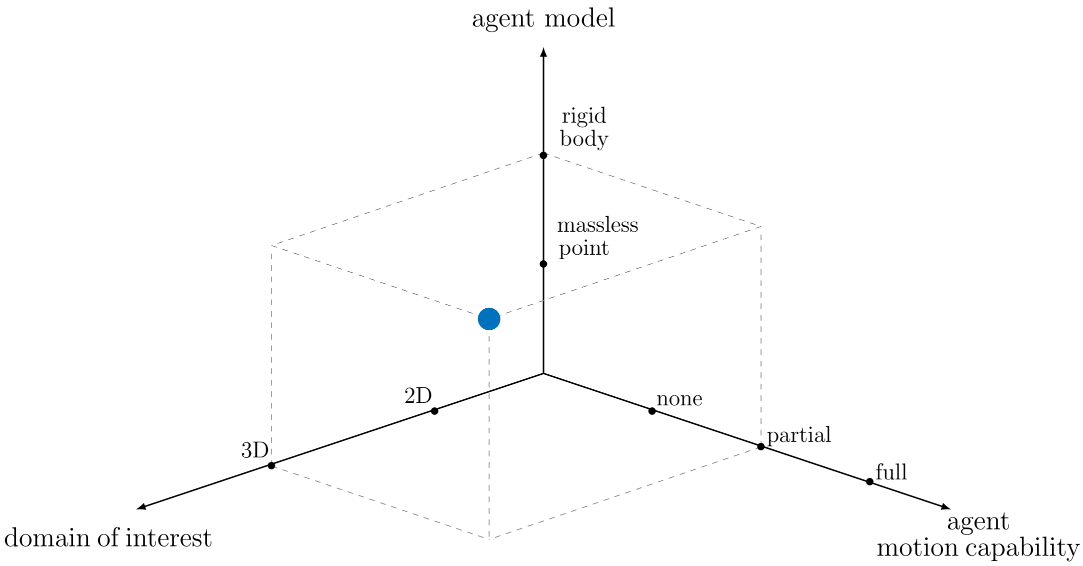


Figure 3.2. VSN: each camera is modeled as a rigid body having partial motion capability and acting in the 3D space.

A camera belonging to a surveillance multi-device system is thus generally modeled as a rigid body having partial motion capability as depicted in Figure 3.2. Its location is, indeed, described by six parameters (three for the position of O_B in the $\mathscr{F}_W$ and three for the orientation of $\mathscr{F}_B$ with respect to $\mathscr{F}_W$), however it can adjust only its attitude having fixed position in the 3D space.

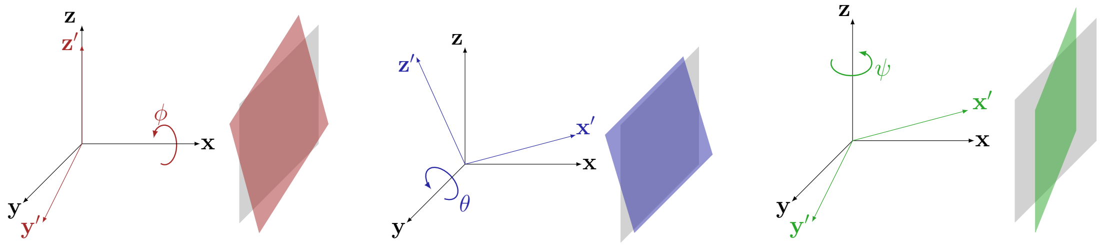
Remark 3.2.1. There exist several ways to represent the attitude of a rigid body acting in the 3D space. The choice of using the rotation matrix convention for the camera is motivated by two main reasons. First, the matrix group $SO(3)$ is a Riemannian manifold with a well-stated mathematical characterization, whose main features are recalled in Appendix B. Then, this representation admits no singularity. According to Euler's rotation theorem (Kuipers et al. (1999)), any orientation may be also identified by the triplet of angles, $(\phi,\ \theta,\ \psi)$, called *Euler angles*, so that any rotation can be decomposed in a sequence of three rotations around the axes identified by the unit vectors $\mathbf{e}_i, i \in \{1, 2, 3\}$ of the canonical basis of $\mathbb{R}^3$. Explicitly, it is

$$\mathbf{R} = \mathbf{R}(\phi, \theta, \psi) = \mathbf{R}^z(\psi)\mathbf{R}^y(\theta)\mathbf{R}^x(\phi), \tag{3.1}$$

where, as shown in Figure 3.3,

- $\mathbf{R}^x(\phi)$ indicates a counterclockwise rotation of $\phi \in (-\pi, \pi]$ (*roll angle*) around $\mathbf{e}_1$,
- $\mathbf{R}^y(\theta)$ points a counterclockwise rotation of $\theta \in (-\pi, \pi]$ (*tilt/pitch angle*) around $\mathbf{e}_2$,
- $\mathbf{R}^z(\psi)$ states a counterclockwise rotation of $\psi \in (-\pi, \pi]$ (*pan/yaw angle*) around $\mathbf{e}_3$.

The Euler angles formalism has the advantage to have a physical interpretation, however it suffers of a singularity point in correspondence to $\theta = \pm\frac{\pi}{2}$. In this point, it takes place the gimbal lock phenomenon which causes the control authority loss on one DoF.



(a) $\mathbf{R}^x(\phi)$ **(b)** $\mathbf{R}^y(\theta)$ **(c)** $\mathbf{R}^z(\psi)$

Figure 3.3. Rotation of an arbitrary reference system and the corresponding image plane according to the Euler angles representation.

Let now consider a camera system composed by n visual devices. For sake of simplicity, hereafter it is assumed that the visibility and communication graphs are coincident and that the agents interactions are bilateral, hence the network can be identified with a unique undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Specifically, the presence of an edge $(v_i, v_j) \in \mathcal{E}$ between nodes $v_i, v_j \in \mathcal{V}$ means that the fields of view (FoVs) of the i-th and the j-th camera are at least partially overlapping and that the two devices v_i and v_j are able to reciprocally exchange information according to some preset communication protocol.

For each pair of cameras v_i and v_j with overlapping FoVs (i.e., such that $(v_i, v_j) \in \mathcal{E}$), the attitude $\mathbf{R}_i$ of the i-th camera is linked to the attitude $\mathbf{R}_j$ of the neighboring j-th camera by the rotation composition $\mathbf{R}_i \circ \mathbf{R}_j = \mathbf{R}_i\mathbf{R}_j$ (Ma et al. (2004)). Consequently, the relative change of coordinates is

$$^{i}\mathbf{R}_j = \mathbf{R}_i^{-1} \circ \mathbf{R}_j = \mathbf{R}_i^{\top}\mathbf{R}_j, \tag{3.2}$$

and consistently, the attitude of camera v_j with respect to that of v_i results as

$$\mathbf{R}_j = \mathbf{R}_i \circ^{i} \mathbf{R}_j = \mathbf{R}_i \, {}^{i}\mathbf{R}_j. \tag{3.3}$$

In the following, $^{i}\mathbf{R}_j$ is referred as the *relative orientation* of the i-th camera with respect to the j-th neighbor, whereas its attitude $\mathbf{R}_i$ in $\mathscr{F}_W$ is said *absolute*. Figure 3.4 aims at clarifying the introduced quantities and their relations.

Exploiting relation (3.2), the attitude estimation problem for the given camera system can be formalized through the following definition inspired by Tron and Vidal (2009).

Definition 3.2.2 (Oriented network)**.** A VSN of n cameras associated with the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be *oriented* if there is a set of relative orientations, namely $\{^{i}\mathbf{R}_j \in SO(3), \forall\ (v_i, v_j) \in \mathcal{E}\}$, such that the absolute orientation $\mathbf{R}_i$ $(i = 1 \ldots n)$ of all the devices in the system is uniquely determined arbitrarily fixing the frame of any agent.

According to Definition 3.2.2, the solution of the attitude estimation problem, interpreted as the retrieval of the absolute orientations with respect to the world frame $\mathscr{F}_W$, is conditional to the knowledge of at least the attitude of a device in the network. This a priori information is rarely available. For this reason, it is a common practise to solve a 'relaxed' version of the attitude estimation problem where the body frame of any camera

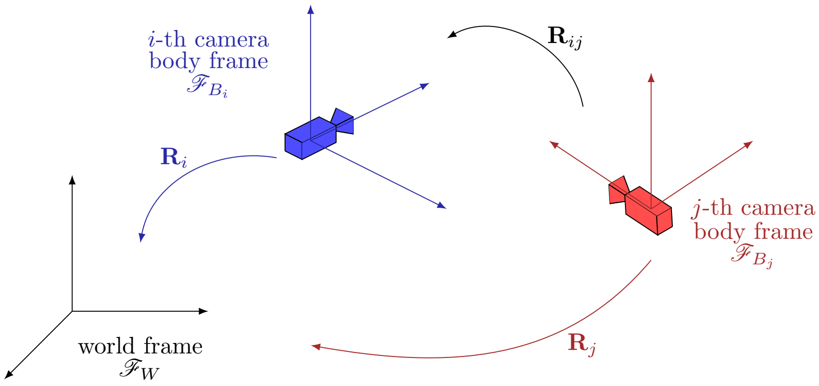


Figure 3.4. Absolute and relative orientation for a camera system where each device is modeled according to the pinhole model.

is fixed as reference frame. In this way, the final estimates are all biased by the same rotation with respect to the inertial frame.

To solve the attitude estimation issue it is also usually assumed that each camera can gather noisy relative orientations measurements with respect to all its neighbors. Such information is routinely provided by means of well-established algorithms that employ only local information about the observed scene as the eight-point algorithm in Hartley (1997) or other variations (see e.g. Chojnacki and Brooks (2003), Ma et al. (2004)).

In this chapter, a set of relative noisy measurements $\widetilde{\mathcal{R}} = \{{}^i\widetilde{\mathbf{R}}_j \in SO(3), \forall\, (v_i, v_j) \in \mathcal{E}\}$ is thus assumed to be available and the solution provided by Tron and Vidal (2009, 2014) for the localization problem is considered restricting to the rotation part. Hereafter, this is referred as *Tron-Vidal solution* (TV solution).

3.3 Tron-Vidal Solution

The main idea of the Tron-Vidal solution (Tron and Vidal (2009, 2014)) is to find the set of relative orientations that satisfy the consistency constraints in Definition 3.2.2 and that, at the same time, are as close as possible to the available measurements through the minimization of a suitable cost function. In detail, the rationale behind such minimization-based approach is a generalization of the classical average consensus procedure that is extended to the case of Riemannian manifold $SO(3)$.

From a practical point of view, the algorithm takes as input the set of the noisy relative measurements $\widetilde{\mathcal{R}}$ between each pair of connected cameras in the graph to iteratively compute a set of relative rotations $\{{}^i\widehat{\mathbf{R}}_j \in SO(3),\ \forall\, (v_i, v_j) \in \mathcal{E}\}$, consistently with the oriented network definition. Hence, exploiting relation (3.3), the absolute orientations $\{\widehat{\mathbf{R}}_i \in SO(3)\}_{i=1}^n$ are estimated for all the devices in the system, with respect to the same reference frame once fixed $\widehat{\mathbf{R}}_r = \mathbf{I}_3$ for any $r \in \{1 \dots n\}$. In detail, the suggested criterion is a least squares approach, wherein the cost function to minimize rests upon the Riemannian metric $d_{SO(3)}(\cdot)$ defined on the $SO(3)$ manifold (see Appendix B for further details).

Formally, the aforementioned cost function accounts for the mismatch between the noisy measurements and the computed estimates resulting to be

$$J\left(\{{}^i\widehat{\mathbf{R}}_j\}\right) = \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{N}_i} \left(\frac{1}{2}\, d^2_{SO(3)}({}^i\widehat{\mathbf{R}}_j, {}^i\,\widetilde{\mathbf{R}}_j) \right). \tag{3.4}$$

To satisfy the consistency constraints of Definition 3.2.2 each relative transformation ${}^i\widehat{\mathbf{R}}_j$ is reparametrized in terms of the absolute orientations $\widehat{\mathbf{R}}_i$ and $\widehat{\mathbf{R}}_j$ according to (3.2). As a consequence, the cost function (3.4) becomes

$$J\left(\{\widehat{\mathbf{R}}_i\}\right) = \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{N}_i} \left(\frac{1}{2}\, d^2_{SO(3)}(\widehat{\mathbf{R}}_i^\top \widehat{\mathbf{R}}_j, {}^i\,\widetilde{\mathbf{R}}_j) \right) \tag{3.5}$$

$$= \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{N}_i} \left(\frac{1}{4} \left\| \log\left(\widehat{\mathbf{R}}_j^\top \widehat{\mathbf{R}}_i^i \widetilde{\mathbf{R}}_j \right) \right\|_F^2 \right) = \sum_{v_i \in \mathcal{V}} J_i(\widehat{\mathbf{R}}_i). \tag{3.6}$$

Because of (3.6), the attitude estimation problem for a camera system can be solved in a distributed way: each agent is required to solve the non-linear minimization of the local cost function $J_i(\widehat{\mathbf{R}}_i)$ depending on its state $\widehat{\mathbf{R}}_i$ and on $\widehat{\mathbf{R}}_j$, ${}^i\widetilde{\mathbf{R}}_j$ with $v_j \in \mathcal{N}_i$, i.e., on information available through 1-hop communication. This optimization can be achieved using the iterative consensus framework. At every iteration, each node v_i of the network computes the Riemannian gradient of $J_i(\widehat{\mathbf{R}}_i)$ with respect to $\widehat{\mathbf{R}}_i$, identifying the geodesic in the tangent space of $\widehat{\mathbf{R}}_i$ along which it needs to update its rotation estimate. All the nodes compute the gradients and the updates at the same time and independently.

In detail, at every iteration t, each node v_i performs the following procedure.

1. Denoting the current estimate of the its absolute orientation $\mathbf{R}_i$ by $\widehat{\mathbf{R}}_i(t)$, it first determines the Riemannian gradient of $J_i(\widehat{\mathbf{R}}_i(t))$ with respect to $\widehat{\mathbf{R}}_i(t)$, namely it computes the matrix $\mathbf{G}_{J_i} = \nabla J_i(\widehat{\mathbf{R}}_i(t)) \in \mathbb{R}^{3\times 3}$ such that

$$\mathbf{G}_{J_i} = \left(\frac{\partial J_i(\widehat{\mathbf{R}}_i(t))}{\partial \widehat{\mathbf{R}}_i(t)} \right) - \widehat{\mathbf{R}}_i(t) \left(\frac{\partial J_i(\widehat{\mathbf{R}}_i(t))}{\partial \widehat{\mathbf{R}}_i(t)} \right)^\top \widehat{\mathbf{R}}_i(t) \tag{3.7}$$

$$= -\widehat{\mathbf{R}}_i(t) \sum_{v_j \in \mathcal{N}_i} \left(\log(\widehat{\mathbf{R}}_j(t)^\top \widehat{\mathbf{R}}_i(t)^i \widetilde{\mathbf{R}}_j^\top) + \log(\widehat{\mathbf{R}}_i(t)^\top \widehat{\mathbf{R}}_j(t)^j \widetilde{\mathbf{R}}_i^\top) \right), \tag{3.8}$$

   where $\frac{\partial J_i(\widehat{\mathbf{R}}_i(t))}{\partial \widehat{\mathbf{R}}_i(t)}$ is the Euclidean derivative of the function $J_i(\widehat{\mathbf{R}}_i(t))$ and $\log(\cdot)$ is the logarithmic map defined over $SO(3)$.

2. It determines the estimation $\widehat{\mathbf{R}}_i(t+1)$ related to the subsequent iteration by performing a gradient descent step. In detail, moving along the geodesic in antigradient direction with a properly chosen step-size $\varepsilon \in \mathbb{R}^+$, $\widehat{\mathbf{R}}_i(t+1)$ is computed as

$$\widehat{\mathbf{R}}_i(t+1) = \exp_{\widehat{\mathbf{R}}_i(t)}(-\varepsilon\, \mathbf{G}_{J_i}), \tag{3.9}$$

   where $\exp_{\widehat{\mathbf{R}}_i(t)}(\cdot)$ is the exponential map at $\widehat{\mathbf{R}}_i(t) \in SO(3)$ (further details about the functions defined over the manifold $SO(3)$ are available in Appendix B). The rotation estimate is then communicated to all the neighboring nodes $v_j \in \mathcal{N}_i$.

The iterative algorithm stops after a pre-defined number $t_{max} \in \mathbb{N}$ of iterations, which has to be large enough to guarantee the achievement of a minimum for the cost function.

3.4 Initialization Methods for TV Solution

The TV minimization-based algorithm (Tron and Vidal (2009, 2014)) leverages on the noisy relative measurements among the cameras in the network and envisages the iterative communication of the estimated absolute rotations among the nodes. Given the non-convexity of the involved cost function (3.4), to allow for the convergence towards a correct estimate, it is therefore necessary to initialize the matrices $\{\widehat{\mathbf{R}}_i(0)\}_{i=1}^n$ at the zero-th iteration with a good guess.

This section is entirely devoted to the presentation of some initialization methods that differ in a priori information requirement, computational load and robustness to measurement noise. The underlying idea shared by all these methods is that it is necessary to extract a subgraph from the VSN graph in order to assign initial values consistently with the oriented network requirement stated by Definition 3.2.2.

3.4.1 Single Spanning Tree Method

The easiest way to design an initialization strategy briefly cited in Tron and Vidal (2009, 2014) is the *single spanning tree method* (SST). As illustrated in Algorithm 3, given a camera network associated to the graph $\mathcal{G}$ and the set of relative measurements $\widetilde{\mathcal{R}}$, the SST procedure consists of three main steps:

1. to choose any node as a root (e.g., node v_1) and to impose $\widehat{\mathbf{R}}_1(0) = \mathbf{I}_3$ (line 2);

2. to find a spanning tree $\mathcal{ST}_{\mathcal{G},1}$ rooted in v_1 (introduced in Section 1.2.2) that provides simple paths ℓ_{1i} from the root node to any other node v_i in the network (line 3);

3. to set $\widehat{\mathbf{R}}_i(0) = \widehat{\mathbf{R}}_1(0)\widetilde{\mathbf{R}}_{\ell_{1i}}^\top$ for all $v_i \in \mathcal{V}$, where $\widetilde{\mathbf{R}}_{\ell_{1i}} =^i \widetilde{\mathbf{R}}_j \circ \cdots \circ^k \widetilde{\mathbf{R}}_1$ is the relative rotations composition along the path $\ell_{1i} = \{v_1, v_k, \dots v_j, v_i\}$ in the designed rooted spanning tree (line 5).

There is a level of arbitrariness in the choice of the root/reference node, which eventually may affect the reconstruction accuracy. In fact, it is important to observe that the estimated absolute orientation (and thus the error with respect to the true value) is obtained through a composition law. Intuitively, the relative transformations errors are likewise composed, so the lack of accuracy increases with the distance of the i-th node from the reference node. Thus, as a guideline, it is advisable to select a spanning tree as balanced as possible, wherein the differences of the paths lengths are as small as possible. A possible strategy to find a balanced spanning tree is to use an algorithm that computes the minimum spanning tree, e.g., Kruskal o Prim algorithm (Cormen, Leiserson, Rivest, and Stein (2009)).

Algorithm 3 SST

```
procedure SST($\mathcal{G}$,$\widetilde{\mathcal{R}}$)
    Set root = 1 and $\widehat{\mathbf{R}}_1 = \mathbf{I}_3$
    Define a spanning tree $\mathcal{ST}_{\mathcal{G},1}$
    for $i \leftarrow 2$ to $n$ do
        Compute $\widehat{\mathbf{R}}_i = \widehat{\mathbf{R}}_1 \widetilde{\mathbf{R}}_{\ell_{1i}}^\top$
    end for
end procedure
```

In summary, the SST initialization method has the advantage of being fast, nevertheless the robustness of the whole algorithm crucially depends on the root node that must be manually selected in a centralized fashion or through a *leader election* procedure.

3.4.2 Multipath Method

A more complex initialization strategy that aims at reducing the arbitrariness in the choice of the root node is the *multipath method* (MP). This consists in the definition of several paths in the network and in the averaging of multiple absolute rotation estimates for each camera. With reference to Algorithm 4, given a camera network associated to the graph $\mathcal{G}$ and the set of relative measurements $\widetilde{\mathcal{R}}$, the procedure starts similarly to SST approach as the root node (e.g., node v_1) is fixed setting $\widehat{\mathbf{R}}_1(0) = \mathbf{I}_3$ (line 2). Then, for each other node v_i, four steps are performed. In detail,

1. all possible m_i paths ℓ_{1i}^k (with $k = 1 \dots m_i$) from the reference node to node v_i are determined (line 4);

2. the rotation estimate $\widehat{\mathbf{R}}_i^k(0) = \widehat{\mathbf{R}}_1(0)\widetilde{\mathbf{R}}_{\ell_{1i}^k}^\top$ is computed using the relative rotation composition rule (3.2) along the k-th path to obtain m_i different estimates (line 6);

3. each k-th estimate is associated to a weight $w_i^k \in \mathbb{R}^+$ equal to the reciprocal of the k-th path length, up to a normalization factor (line 7), i.e.,

$$w_i^k = \frac{1}{|\ell_{1i}^k|} \frac{1}{\sum_{k=1}^{m_i} \frac{1}{|\ell_{1i}^k|}}; \tag{3.10}$$

4. the final rotation estimate $\widehat{\mathbf{R}}_i(0)$ is thus derived as the mean of $\{\widehat{\mathbf{R}}_i^1(0), \dots \widehat{\mathbf{R}}_i^{m_i}(0)\}$ weighted by $\{w_i^1, \dots, w_i^{m_i}\}$ according to Algorithm 5 in Appendix B (line 9).

The main advantage of MP is that the uncertainty on the initial absolute rotation is generally reduced for two reasons. First, by averaging on different rotations, a priori information about the relative transformations is better exploited. Second, the adoption of the weighting factors (3.10) allows to mainly consider the estimates computed using the shortest paths, which provide less noisy estimates, for each device. On the other hand, the computational burden can in principle become prohibitive because of the calculation of the rotations mean and the identification of all the paths connecting two nodes (which is known to be a NP-hard problem).

Algorithm 4 MP

```
procedure MP($\mathcal{G}$,$\widetilde{\mathcal{R}}$)
    Set root = 1 and $\widehat{\mathbf{R}}_1 = \mathbf{I}_3$
    for $i \leftarrow 2$ to $N$ do
        Compute all possible $\ell_{1i}^k$, $k = 1, \dots, m_i$
        for $k \leftarrow 1$ to $m_i$ do
            Compute $\widehat{\mathbf{R}}_i^k = \widehat{\mathbf{R}}_1 \widetilde{\mathbf{R}}_{\ell_{1i}^k}^\top$
            Compute $w_i^k = \frac{1}{|\ell_{1i}^k|} \frac{1}{\sum_{k=1}^{m_i} \frac{1}{|\ell_{1i}^k|}}$
        end for
        Compute the weighted mean of $\{\widehat{\mathbf{R}}_i^1, \dots \widehat{\mathbf{R}}_i^{m_i}\}$
    end for
end procedure
```

To avoid this latter issue, only a subset of all the possible paths is evaluated, for instance the subset of paths having fixed length $l \ll n$. By suitably selecting the parameter l according to the network topology, the benefits of MP are preserved, while the growth of the overall computational complexity is controlled.

3.4.3 Averaged versions

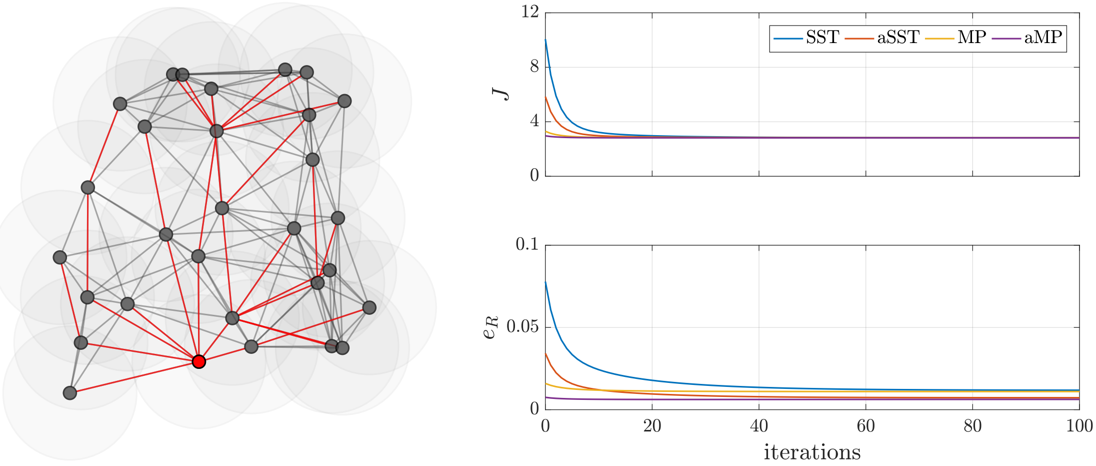
Both SST and MP can be slightly modified in order to better exploit the input measurements. More precisely, being v_i a neighbor of v_j in the camera network, the fact that ${}^i\widetilde{\mathbf{R}}_j$ is generally different from ${}^j\widetilde{\mathbf{R}}_i$ because of the stochastic distribution of measurement noise, suggests to compute the rotation estimate of the j-th camera in two ways. Indeed, $\widehat{\mathbf{R}}_j$ can be obtained either using the relative rotation ${}^i\widetilde{\mathbf{R}}_j$ or employing ${}^j\widetilde{\mathbf{R}}_i$, namely $\widehat{\mathbf{R}}_j' = \widehat{\mathbf{R}}_i\,{}^i\widetilde{\mathbf{R}}_j$ and $\widehat{\mathbf{R}}_j'' = \widehat{\mathbf{R}}_i\,{}^j\widetilde{\mathbf{R}}_i^\top$. By averaging the two relative transformations, new initialization procedures, called *averaged single spanning tree method* (aSST) and *averaged multipath method* (aMP), can be introduced: in aSST and aMP the relative rotation ${}^i\widetilde{\mathbf{R}}_j$ is replaced by the (simple) mean of ${}^i\widetilde{\mathbf{R}}_j$ and ${}^j\widetilde{\mathbf{R}}_i^\top$ (see Appendix B).

3.4.4 Numerical and Experimental Results

This section is devoted to the validation and comparison of the designed SST, MP, aSST, and aMP algorithms, by adopting them as the initialization step for the Riemannian consensus-based approach proposed by the TV solution described in Section 3.3. To this aim, both a synthetic and a real scenarios are considered for a thorough analysis. Furthermore, in order to provide a measure of the goodness of the orientation estimate, it is useful to introduce the *mean error* on rotations at the t-th iteration, $e_R(t)$, defined as

$$e_R(t) = \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{R}_i - \widehat{\mathbf{R}}_i(t)\|_F^2, \tag{3.11}$$

where n is the number of cameras in the network, $\mathbf{R}_i$ denotes the real absolute orientation of the i-th camera, $\widehat{\mathbf{R}}_i(t)$ indicates its estimate at t-th iteration of the algorithm.



(a) Network+ST **(b)** Cost function+Mean error

Figure 3.5. Numerical scenario: (a) 30-nodes random geometric graph and spanning tree used in SST method (in red); (b-top) cost function and (b-bottom) mean error on absolute rotations, averaging over 100 realizations of ${}^i\widetilde{\mathbf{R}}_j =^i \mathbf{R}_j\, {}^i\mathbf{R}_j^n(\phi,\theta,\psi)$, $\phi,\theta,\psi \sim \mathcal{N}(0^\circ,5^\circ)$.

	SST	aSST	MP	aMP
$e_R(0)$	0.092	0.047	0.019	0.009
$e_R(t_{max})$	0.017	0.010	0.012	0.008
$\lvert e_R(t_{max}) - e_R(0)\rvert$	0.075	0.037	0.007	0.001

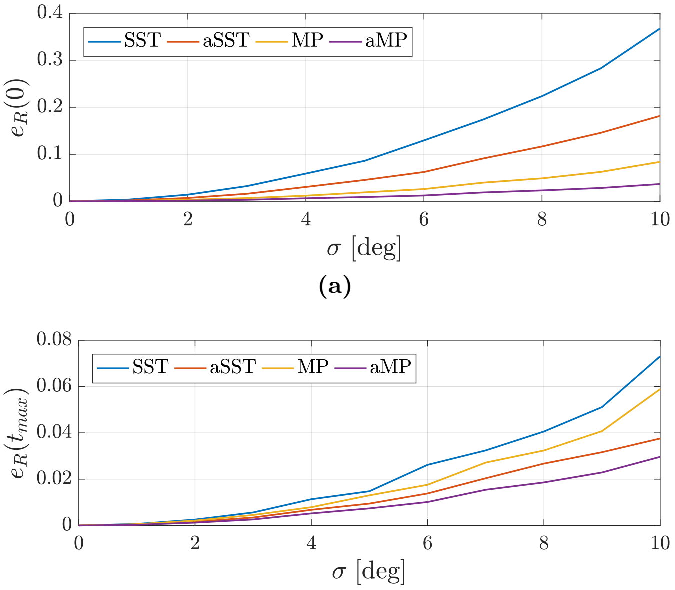
Table 3.1. Comparison among different initialization strategies on the numerical dataset.

Numerical Validation

The numerical scenario considers a networked system made up of $n = 30$ omnidirectional cameras connected in a geometric random graph. This network is displayed in Figure 3.5a: the gray circles are the field of view of the cameras and the red lines depict the considered spanning tree rooted in the red node. As far as MP and aMP are concerned, only paths with maximum length equal to $l = 5$ are computed.

The noisy relative rotation ${}^i\widetilde{\mathbf{R}}_j$ between each pair of neigboring cameras is obtained by composing the exact relative measurement with a noise term, namely a rotation matrix ${}^i\mathbf{R}_j^n$ whose pan-tilt-roll angles (ϕ, θ, ψ in (3.1)) are randomly selected according to a Gaussian distribution with standard deviation $\sigma = 5^\circ$. The minimization of the cost function J is performed through $t_{max} = 100$ iterations imposing the step-size ε to 0.01.

The different initialization methods are compared by evaluating the mean error (3.11) on the estimates obtained at the end of the algorithm execution, $e_R(t_{max})$, as opposed to the initial one that results from the initialization phase, $e_R(0)$. The mean error trend is reported in Figure 3.5b-bottom. Similarly, Figure 3.5b-top shows the trend of the cost function (3.4), and verifies that the minimization of J is attained in all cases. In this sense, it is necessary to clarify that the cost function is related to the relative rotations, while the rotation errors are computed on the absolute ones.



(a)

(b)

Figure 3.6. Numerical scenario: initial (a) and final (b) mean error on the estimated absolute rotations for different values of the measurements noise standard deviation, $\sigma \in [0°, 10°]$.

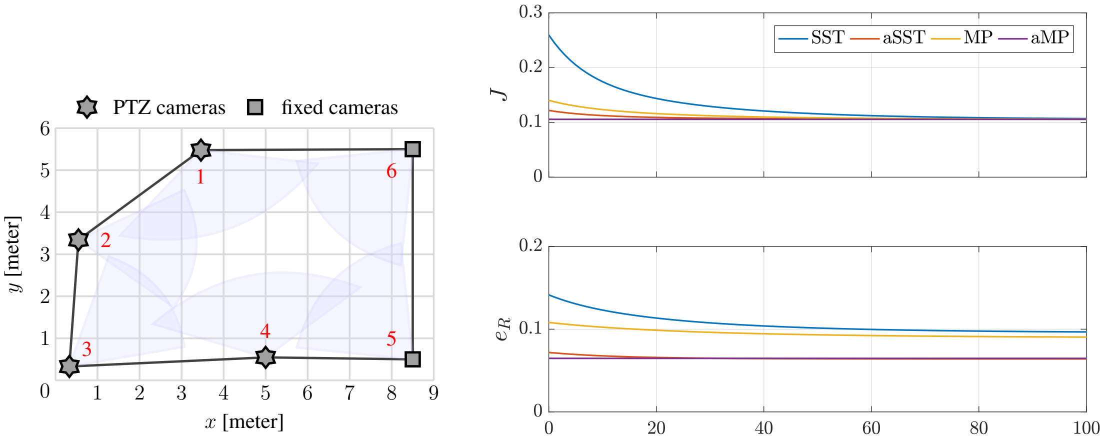
Table 3.1 reports the error values obtained averaging the results for 100 noise realizations of the same dataset setting. The exploitation of the a priori information about the relative transformations allows to achieve better estimates to initialize the Tron-Vidal iterative procedure. Specifically, the smallest error $e_R(0)$ is obtained using the aMP strategy, and analogously for $e_R(t_{max})$. Note that the aMP method by itself (i.e., $e_R(0)$ for aMP) returns estimates that are very close to the optimal ones, i.e., those computed by the complete algorithm ($e_R(t_{max})$ for aMP). Nevertheless, a similar performance can be achieved also by initializing the iterative procedure with the aSST algorithm ($e_R(t_{max})$ for aSST), which is computationally simpler.

It is also interesting to study how the different methods perform with an increasing level of noise. Figure 3.6 reports the behavior of the mean estimation error with noise standard deviation in the range $\sigma \in [0°, 10°]$. Clearly, the estimation error generally increases with the measurement noise, although the averaged approach aSST and aMP allow to reach lower final estimation errors. Furthermore, the performance of aSST method is characterized also by a good error reduction with respect to the initial value.

Experiments on a Real Scenario

To better provide an insight over the proposed algorithms, some experimental data have been collected in the real camera network setup available in the NAVLAB laboratory[2]. This camera system consists of both fixed low-cost cameras and Pan-Tilt-Zoom (PTZ) cameras, which can vary their orientation on their y-axis and z-axis keeping fixed the roll angle. The setup configuration is shown in Figure 3.7a.

[2] NAVLAB - Autonomous Navigation Laboratory, Department of Information Engineering, University of Padova, https://sparcs.dei.unipd.it/index.php/laboratories.



(a) Setup (top view)

(b) Cost function+Mean error

Figure 3.7. Real scenario: (a) experimental setup; (b-top) cost function and (b-bottom) mean error on absolute rotations.

	SST	aSST	MP	aMP
$e_R(0)$	0.145	0.072	0.108	0.065
$e_R(t_{max})$	0.097	0.064	0.090	0.065
$\lvert e_R(t_{max}) - e_R(0)\rvert$	0.048	0.008	0.018	0.000

Table 3.2. Comparison among different initialization strategies on the real dataset.

To mimic a typical perimeter patrolling scenario, the network topology is represented by a circulant graph where, due to the physical constraints of limited FoV, each camera shares visibility region and can communicate only with its two nearest neighbors (namely, a circulant graph with $j = 1$ according to the description provided in Section 1.2.2). The set of the noisy relative poses are derived using the Bouguet calibration toolbox (Bouguet (2000)), starting from a set of 2D images, one for each pair of adjacent devices.

The performances of the four different strategies illustrated in Section 3.4 are compared, by choosing camera 1 in Figure 3.7a as the root node and by selecting a 'balanced' spanning tree consisting in paths 1-2-3-4 and 1-6-5 when the SST methods is taken into account. Moreover, due to the simple ring topology and the limited number of the agents, all possible paths are considered in the MP strategy. The results in terms of mean reconstruction error are shown in Table 3.2 where the initial condition for the Tron-Vidal iterative minimization algorithm is compared with respect to its final result.

Consistently with what observed in the numerical scenario, even using real measurements the best estimates are achieved by employing the averaged methods, aSST and aMP, due to the presence of noise that may strongly affect the measurements. Most remarkably, in the case of aMP, the initial estimates are slightly better than the final rotation estimates calculated by applying the minimization-based algorithm, as well depicted in Figure 3.7b-bottom. This fact suggests that aMP algorithm may be beneficially exploited per se as an alternative and efficient centralized algorithm for camera orientation task in real networks.

Remark 3.4.1. A trade-off is posed between the error of the estimation procedure and its computational cost: in practice, one could either adopt iterative solutions with the proposed initialization or rely on the direct employment of these algorithms with no further iteration, according to the measurement noise levels and the complexity of the network structure. As a guideline, it can be stated that when the network complexity is high, the single spanning tree solutions may be preferable in conjunction with the iterative scheme, whereas, when the measurement noise increases, the averaging algorithms aSST and aMP provide a good performance also when employed in a non-iterative fashion.

As a final observation, by comparing the numerical and the experimental dataset results, it can be noted how the realistic distribution of noise affects the performance of the rotation estimation: in the specific considered multi-camera setup (Figure 3.7a) the relative rotations more prone to errors are those related to fixed low-cost cameras, since their image resolution is lower than that of PTZ cameras. Hence, adopting the averaging procedures in presence of real unbalanced noise distribution appears particularly beneficial. On the other hand, comparing Figure 3.5b-top and Figure 3.7b-top, it can be observed that all the cost function behaviors are similar although the initial and convergence values are different.

3.5 Planar Case

In general, for camera systems used in robotic applications, as well as for omnidirectional camera surveillance networks, the attitude estimation problem is defined on the $SO(3)$ manifold and regards the estimation of three DoFs.

In fixed installations, however, the employment of only PTZ cameras limits the DoFs in the rotational positioning to the angle of yaw (pan) and pitch (tilt), making the camera image plane aligned with respect to the horizon. In this situation, the attitude of any i-th device is given by $\mathbf{R}_i = \mathbf{R}^z(\psi_i)\mathbf{R}^y(\theta_i)\mathbf{I}_3 \in SO(3)$; hence the attitude estimation problem can be recast into an optimization problem over the bi-dimensional unitary group $\mathbb{S}^2 = \{\mathbf{x} \in \mathbb{R}^3 \mid \|\mathbf{x}\|_2 = 1\}$. This represents the sphere embedded in $\mathbb{R}^3$ whose points are identified by two angles.

A further simplification is straightforward in perimeter-patrolling applications where the employed camera system is usually composed of an arbitrary number of devices designed to be able to move backward and forward along their patrolling range (a more detailed explanation of this scenario will be provided in the next chapter where the perimeter patrolling problem is faced). In this framework, each device is modeled as a rigid body having partial motion capability limited to the variation of only one angle (generally, the pan angle). It will be shown that this situation boils down to a 2D scenario, as specified by Figure 3.8, where the agents have only one controllable DoF.

From a mathematical perspective, indeed, when only the rotations around the z-axis are allowed, the absolute and the relative rotation matrices of an arbitrary camera v_i belonging to the neighboring of the camera v_j, namely $\mathbf{R}_i$ and ${}^i\mathbf{R}_j$ in $SO(3)$ respectively,

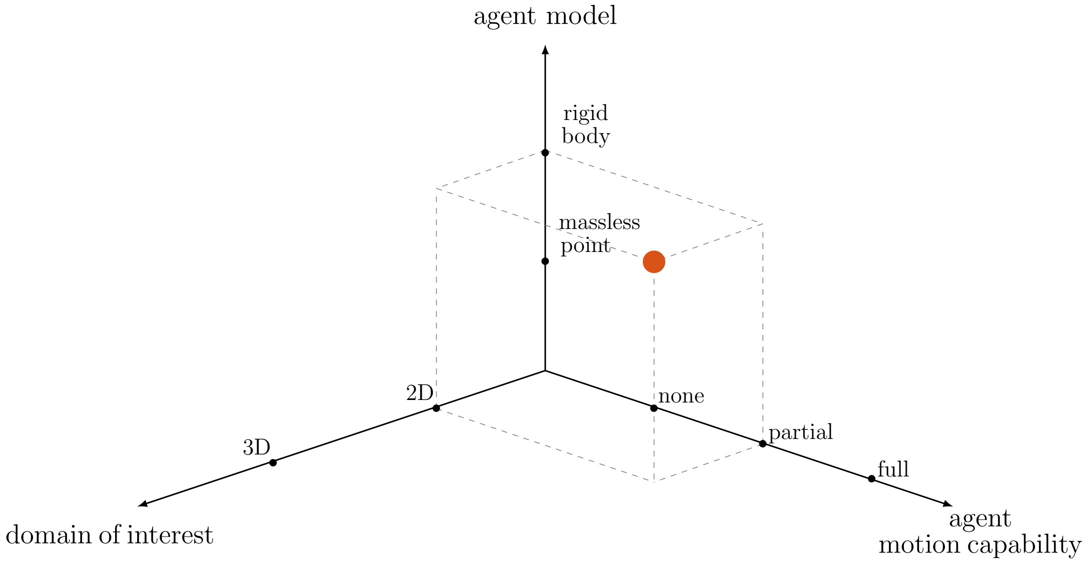


Figure 3.8. planar VSN: each camera is modeled as a rigid body having partial motion capability and acting in 2D space.

result to be

$$\mathbf{R}_i = \begin{bmatrix} \cos\psi_i & -\sin\psi_i & 0 \\ \sin\psi_i & \cos\psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad {}^i\mathbf{R}_j = \begin{bmatrix} \cos{}^i\psi_j & -\sin{}^i\psi_j & 0 \\ \sin{}^i\psi_j & \cos{}^i\psi_j & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.12}$$

with ${}^i\psi_j = \psi_j - \psi_i \in (-\pi, \pi]$. Hence, the attitude estimation task can be redefined as an *angular synchronization problem* over the circle group $\mathbb{S}^1 = \{z \in \mathbb{C} \mid |z| = 1\}$. As explained in Appendix B, $\mathbb{S}^1$ is isomorphic to the two-dimensional Special Orthogonal group $SO(2) = \{\mathbf{R} \in \mathbb{R}^2 \mid \mathbf{R}^\top\mathbf{R} = \mathbf{I}_2,\ \det(\mathbf{R}) = +1\}$ which describes the rigid body rotations on a plane. Thus, each rotation matrix $\mathbf{R} \in SO(2)$ is biunivocally related to a certain angle $\psi \in \mathbb{S}^1$ whose estimation means univocally determining the planar camera orientation.

3.5.1 Centralized vs Distributed Solution

Adopting the TV-solution under the given premises, the cost function (3.6) can be simplified as follows

$$J\left(\{\widehat{\psi}_i\}\right) = \sum_{v_i\in\mathcal{V}}\sum_{v_j\in\mathcal{N}_i}\left(\frac{1}{2}\left(\widehat{\psi}_i - \widehat{\psi}_j +^i \widetilde{\psi}_j\right)^2\right) = \sum_{v_i\in\mathcal{V}} J_i\left(\widehat{\psi}_i\right), \tag{3.13}$$

where $\widehat{\psi}_i$ and $\widehat{\psi}_j$ are the attitude estimation for the i-th and j-th camera respectively, while $\{{}^i\widetilde{\psi}_j\}$ is the noisy relative measurements among them. Note that the matrices product $\widehat{\mathbf{R}}_j^\top\widehat{\mathbf{R}}_i^i\widetilde{\mathbf{R}}_j$ in (3.6) has boiled down to a rotation associated to the angle $\widehat{\psi}_i - \widehat{\psi}_j +^i \widetilde{\psi}_j$ according to the definition of the principal logarithm in $SO(2)$ reported in Appendix B.

The optimal estimates of the cameras attitude, thus, result from the solution of the following optimization problem

$$\left\{\widehat{\psi}_1 \dots \widehat{\psi}_n\right\} = \arg \min_{\psi_1 \dots \psi_n} \; J\left(\{\widehat{\psi}_i\}\right) \tag{3.14}$$

subject to the constraints $\widehat{\psi}_i \in (-\pi, +\pi],\; i = 1 \dots n$. For this purpose, it is worth to notice that the gradient $\mathbf{g}_J = \nabla J\left(\{\widehat{\psi}_i\}\right) \in \mathbb{R}^n$ and the Hessian $\mathbf{H}_J = \nabla^2 J\left(\{\widehat{\psi}_i\}\right) \in \mathbb{R}^{n \times n}$ of the cost function (3.13) are such that

$$[\mathbf{g}_J]_i = 2\deg(v_i)\widehat{\psi}_i - 2\sum_{v_j \in \mathcal{N}_i} \widehat{\psi}_j - \sum_{v_j \in \mathcal{N}_i} ({}^j\widetilde{\psi}_i -{}^i \widetilde{\psi}_j) \tag{3.15}$$

$$[\mathbf{H}_J]_{ij} = \begin{cases} 2\deg(v_i) & \text{if} \quad i = j \\ -2 & \text{if} \quad i \neq j, \quad v_j \in \mathcal{N}_i \\ 0 & \text{if} \quad i \neq j, \quad v_j \notin \mathcal{N}_i \end{cases} \tag{3.16}$$

Note that the second derivative of $J\left(\{\widehat{\psi}_i\}\right)$ results to be equal to twice the Laplacian matrix $\mathbf{L}$ of the graph $\mathcal{G}$, namely $\mathbf{H}_J = 2\mathbf{L}$, and that in the considered 2D scenario, J is a convex function over the convex set defined by $\{\psi_i \in (-\pi, \pi]\}$, thus the local minimum of the optimization problem (3.14) is also a global minimum.

As a matter of fact, imposing $\mathbf{g}_J = \mathbf{0}$ the following equations system is obtained

$$2\mathbf{L} \begin{bmatrix} \widehat{\psi}_1 \\ \vdots \\ \widehat{\psi}_n \end{bmatrix} = \begin{bmatrix} \sum_{v_j \in \mathcal{N}_1} ({}^j\widetilde{\psi}_1 -{}^1 \widetilde{\psi}_j) \\ \vdots \\ \sum_{v_j \in \mathcal{N}_n} ({}^j\widetilde{\psi}_n -{}^n \widetilde{\psi}_j) \end{bmatrix} \tag{3.17}$$

Denoting with $\widetilde{\boldsymbol{\psi}} \in \mathbb{R}^n$ the rhs of (3.17) which depends on the available measurements set and introducing the state vector $\widehat{\boldsymbol{\psi}} = \begin{bmatrix} \widehat{\psi}_1 & \dots & \widehat{\psi}_n \end{bmatrix}^\top \in \mathbb{R}^n$, the following lemma can be straightforwardly stated.

Lemma 3.5.1. *In a planar multi-camera network associated to the graph $\mathcal{G}$, if the relative noisy measurements are such that $\widetilde{\boldsymbol{\psi}} \notin \ker(\mathbf{L})$, the optimal solution to (3.14) is given by*

$$\widehat{\boldsymbol{\psi}}^* = \frac{1}{2}\mathbf{L}^+\widetilde{\boldsymbol{\psi}}, \tag{3.18}$$

where $\mathbf{L}^+ = (\mathbf{L}^\top\mathbf{L})^{-1}\mathbf{L} \in \mathbb{R}^{n \times n}$ is the pseudo-inverse of the graph Laplacian matrix.

In other words, (3.18) represents the *centralized* solution to the attitude estimation problem in the planar case, which requires to know the topology of the network and the noisy relative measurements.

Note that each equation $[\mathbf{g}_J]_i = \mathbf{0}$, i.e., each row of the matrices involved in (3.17), refers to a single camera node information, in terms of local topology (neighborhood $\mathcal{N}_i$) and measurements (${}^i\widetilde{\psi}_j$ and ${}^j\widetilde{\psi}_i$, $\forall v_j \in \mathcal{N}_i$), thus suggesting a distributed approach. More formally, from $[\mathbf{g}_J]_i = \mathbf{0}$ a relation for $\widehat{\psi}_i$ can be written as

$$\widehat{\psi}_i = \frac{\sum_{v_j \in \mathcal{N}_i} \widehat{\psi}_j}{\deg(v_i)} + \frac{\sum_{v_j \in \mathcal{N}_i} ({}^j\widetilde{\psi}_i -{}^i \widetilde{\psi}_j)}{2\deg(v_i)}, \tag{3.19}$$

i.e., the i-th node (camera) updates its state (rotation angle estimate) according to the neighboring nodes $\mathcal{N}_i$ and the set of relative measurements involving itself.

From (3.19) two state-space models for the whole camera system can be derived.

- *State-space model 1*

  By considering the whole state vector $\widehat{\boldsymbol{\psi}} \in \mathbb{R}^n$, a discrete time system of the form

  $$\widehat{\boldsymbol{\psi}}(t+1) = \mathbf{F}\widehat{\boldsymbol{\psi}}(t) + \mathbf{u}, \tag{3.20}$$

  can be written from (3.19) as an update rule driven by the input measurements. In model (3.20) every i-th state at the t-th step affects the neighbors estimates at the $(t+1)$-th step, but it is not considered for the recursive self-estimate. As a consequence, the state matrix $\mathbf{F} \in \mathbb{R}^{n\times n}$ is equal to the adjacency matrix normalized by the node degrees and it is thus a row-stochastic matrix. Formally, it occurs

  $$f_{ij} = \frac{a_{ij}}{\deg(v_i)} \qquad \Rightarrow \qquad \mathbf{F} = \mathbf{D}^{-1}\mathbf{A}$$

  Analogously, the input $\mathbf{u} \in \mathbb{R}^n$ in (3.20) is given by the vector of the normalized relative measurements

  $$\mathbf{u} = \frac{1}{2}\mathbf{D}^{-1}\widetilde{\boldsymbol{\psi}}. \tag{3.21}$$

  From Landau and Odlyzko (1981), if the graph $\mathcal{G}$ representing the given camera system is connected, then the state matrix $\mathbf{F}$ has rank $n-1$ or equivalently it has n real eigenvalues $\lambda_0 \dots \lambda_{n-1}$ in the range $[-1, 1]$ with $\lambda_0 = 1$ having single algebraic multiplicity.

- *State-space model 2*

  A different model can be constructed by adopting (3.19) to provide only a weighted correction to the current estimate, thus leading to

  $$\begin{aligned}\widehat{\boldsymbol{\psi}}(k+1) &= \eta\widehat{\boldsymbol{\psi}}(k) + (1-\eta)\left(\mathbf{F}\widehat{\boldsymbol{\psi}}(k) + \mathbf{u}\right) &(3.22)\\ &= (\eta\mathbf{I}_n + (1-\eta)\mathbf{F})\,\widehat{\boldsymbol{\psi}}(k) + (1-\eta)\mathbf{u} &(3.23)\\ &= \mathbf{F}'(\eta)\widehat{\boldsymbol{\psi}}(k) + (1-\eta)\mathbf{u}, &(3.24)\end{aligned}$$

  where $\mathbf{u}$ is as in (3.21) and $\mathbf{F}'(\eta) \in \mathbb{R}^{n\times n}$ is still row-stochastic but with eigenvalues in the range $[-1+2\eta, 1]$, $\eta \in (0, 1)$.

Note that if the equilibrium points of systems (3.20) and (3.24) exist, these are those stated by Lemma 3.5.1. Indeed, (3.20) converges to (3.18), since at the equilibrium it is

$$\begin{aligned}&\widehat{\boldsymbol{\psi}}^* = \mathbf{F}\widehat{\boldsymbol{\psi}}^* + \mathbf{u} = \mathbf{D}^{-1}\mathbf{A}\widehat{\boldsymbol{\psi}}^* + \frac{1}{2}\mathbf{D}^{-1}\widetilde{\boldsymbol{\psi}} &(3.25)\\ \Rightarrow\quad &(\mathbf{D}-\mathbf{A})\,\widehat{\boldsymbol{\psi}}^* = \frac{1}{2}\widetilde{\boldsymbol{\psi}} &(3.26)\\ \Rightarrow\quad &\widehat{\boldsymbol{\psi}}^* = \frac{1}{2}\mathbf{L}^+\widetilde{\boldsymbol{\psi}} &(3.27)\end{aligned}$$

On the other side, also the solution of (3.24) converges to the centralized one (3.18):

$$\widehat{\boldsymbol{\psi}}^* = \eta\widehat{\boldsymbol{\psi}}^* + (1-\eta)\mathbf{F}\widehat{\boldsymbol{\psi}}^* + (1-\eta)\mathbf{u} = \eta\widehat{\boldsymbol{\psi}}^* + (1-\eta)\mathbf{D}^{-1}\mathbf{A}\widehat{\boldsymbol{\psi}}^* + \frac{1}{2}(1-\eta)\mathbf{D}^{-1}\widetilde{\boldsymbol{\psi}} \tag{3.28}$$

$$\Rightarrow \quad (\mathbf{D}-\mathbf{A})\,\widehat{\boldsymbol{\psi}}^* = \frac{1}{2}\widetilde{\boldsymbol{\psi}} \tag{3.29}$$

$$\Rightarrow \quad \widehat{\boldsymbol{\psi}}^* = \frac{1}{2}\mathbf{L}^+\widetilde{\boldsymbol{\psi}} \tag{3.30}$$

As a consequence, the equilibria of systems (3.20) and (3.24) represent the *distributed* solutions of the attitude estimation problem (3.14).

3.5.2 Convergence Analysis

For the convergence analysis, it is necessary to distinguish between the convergence of the cost function (3.13) towards zero and the convergence of the pan angle estimates to the equilibrium values, being the first a linear combination of the second ones.

It is desirable that the update schemes (3.20) and (3.24) guarantee a monotonic decrease of the cost function, for example by adopting a gradient descent iterative procedure. Nevertheless, the convergence of the angles towards the equilibrium configuration depends on the eigenvalues of the state matrix of the evaluated model ($\mathbf{F}$ or $\mathbf{F}'(\eta)$) and, in particular, the convergence rate depends on its second largest eigenvalue in modulus.

State-space model 1

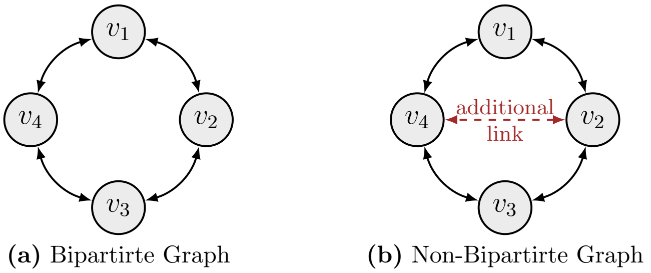
Considering system (3.20) whose dynamics is governed by the matrix $\mathbf{F}$, the angles convergence is ensured only if $\lambda = -1$ is not among the eigenvalues of this state matrix. In fact, if $\lambda = -1$ belongs to the spectrum of $\mathbf{F}$ its multiplicity would be unitary (Landau and Odlyzko (1981)) and this would imply constant oscillations of the angles estimates around their equilibrium values for large observation times ($t \gg 1$) due to the presence of a dominant oscillatory mode.

From a graphical model point of view, this situation can be interpreted with reference to the network topology since $\lambda = -1$ occurs if and only if the graph associated to $\mathbf{F}$ is *bipartite*, i.e., all its cycles consist of an even number of nodes (Landau and Odlyzko (1981)). This statement is formally proved in the following.

Theorem 3.5.2. *Let $\mathbf{F} \in \mathbb{R}^{n\times n}$ be a row-stochastic, non-negative matrix such that its influence graph*[3] *$\mathcal{G}_I(\mathbf{F})$ is strongly connected. Then $\lambda = -1$ is a simple eigenvalue of $\mathbf{F}$ if and only if $\mathcal{G}_I(\mathbf{F})$ is a bipartite graph.*

Proof. Since matrix $\mathbf{F}$ is row-stochastic, $\lambda_0 = 1$ is its largest eigenvalue, while all other eigenvalues have absolute value smaller or equal to 1. Moreover, being its influence graph strongly connected, $\mathbf{F}$ is also irreducible, i.e., it is not similar via a permutation to a

[3] The *influence graph* $\mathcal{G}_I(\mathbf{F})$ of a matrix $\mathbf{F} \in \mathbb{R}^{n\times n}$ is a n-nodes directed graph where there exists a link between nodes v_i and v_j if $[\mathbf{F}]_{ij} \neq 0$.



(a) Bipartirte Graph **(b)** Non-Bipartirte Graph

Figure 3.9. Simple examples of network topology: adding ad edge to a 4-nodes circulant graph (a), a non-bipartite graph is obtained (b).

block upper triangular matrix. The irreducibility and row-stochasticity properties imply that all the eigenvalues of $\mathbf{F}$ with unitary absolute value can be expressed as

$$\lambda_i = e^{j\frac{2\pi k}{\mu}}, \quad k = 0, \ldots, \mu - 1, \tag{3.31}$$

where μ is the so-called *index of imprimitivity*. Furthermore, since $\mathbf{F}$ is non-negative, its index of imprimitivity μ coincides with the maximum common divisor (MCD) of the cycles length in its influence graph (Farina and Rinaldi (2011)).

Given these premises, if the matrix $\mathbf{F}$ has the eigenvalue $\lambda = -1$ then its index of imprimitivity is necessarily even and therefore all the cycles in its influence graph are even-length. As a consequence, the graph $\mathcal{G}_I(\mathbf{F})$ is bipartite.

Conversely, if $\mathcal{G}_I(\mathbf{F})$ is a bipartite graph then the MCD of its cycle lengths is even. As the MCD coincides with the number μ of the peripheral eigenvalues of $\mathbf{F}$, the eigenvalue $\lambda = -1$ occurs in correspondence to $k = \frac{\mu}{2}$ in (3.31). □

Corollary 3.5.3. *Let $\mathbf{F} \in \mathbb{R}^{n\times n}$ be the state matrix of system* (3.20)*, related to the connected graph $\mathcal{G}$. Then $\lambda = -1$ is a simple eigenvalue of $\mathbf{F}$ if and only if $\mathcal{G}$ is a bipartite graph.*

Proof. The graph $\mathcal{G}$ can be considered as the weighted graph corresponding to the adjacency matrix $\mathbf{A}$ where weights are defined by the degree matrix $\mathbf{D}$. Because $\mathbf{A}$ is symmetric, its influence graph can be constructed substituting each undirected edge in $\mathcal{G}$ with a pair of direct arcs having opposed orientation. The obtained influence graph $\mathcal{G}_I$ coincides with the influence graph of the matrix $\mathbf{F}$, because $\mathbf{F} = \mathbf{D}^{-1}\mathbf{A}$.

Given that $\mathcal{G}_I$ is bipartite and strongly connected if and only if $\mathcal{G}$ is bipartite and connected, the thesis of the corollary is proved by Theorem 3.5.2. □

When the graph describing the network is bipartite, the convergence to the equilibrium can be reached only if it is possible to insert a link so that the resulting graph presents at least one cycle made of an odd number of nodes (see Figure 3.9). In practice, in the multi-camera network scenario, this case translates into the possibility of finding an additional connection among cameras that are neighbors in the sense of both communication and visibility, nevertheless such an edge selection solution is barely feasible in real-world obstacle-rich applications.

State-space model 2

The presence of self-loops in the model (3.24) controlled by parameter η allows to modify the eigenvalues domain from the unit circle to the circle centered in $(\eta, 0)$ having radius $\rho = 1 - \eta,\ \rho \in (0, 1)$, ruling out the possible presence of the critic eigenvalue $\lambda = -1$. Therefore, not only the stability of the system is obtained but also the convergence of the pan angle estimates is always assured.

In detail, the η parameter can be tuned in order to control the convergence rate, governed by the second largest eigenvalue (in modulus) of state matrix $\mathbf{F}'(\eta)$. If this is negative the estimated angles converge toward the equilibrium values through an oscillatory transient period. On the contrary, if the second largest eigenvalue is positive then the estimation trend is asymptotically convergent, monotonic for large observation times ($t \gg 1$). From an applicative perspective, the former oscillatory behavior is preferable since an averaging operation might provide an approximate solution to the convergence value, while in the latter case the iterations might consistently underestimate or overestimate the asymptotic values. Remarkably, these different behaviors can be seen as dependent on the control parameter η.

3.5.3 Case Study: a Perimeter-Patrolling Camera Network

Let consider the case of a perimeter-patrolling multi-camera network, namely a system of n visual sensors deployed to surround the scene of interest and so that each camera shares its FoV and can communicate with the preceding and the next devices as in the experimental setup evaluated in Section 3.4.4. Such a topology is described by a circulant graph with a single cycle, hereafter denoted as $\mathcal{C}_n^1$. The choice of this network allows to clarify the theoretical observations about the convergence as the eigenvalues of the state matrix can be expressed in closed form.

Dealing with the state-space model 1, the spectrum of the state matrix $\mathbf{F}$ has the analytic expression

$$\Lambda(\mathbf{F}) = \left\{ \lambda_i \in \mathbb{R} \mid \lambda_i = \cos\left(\frac{2\pi}{n} i\right),\ i = 0 \dots n-1 \right\}. \tag{3.32}$$

Therefore, as the eigenvalues are real, $\lambda = -1$ belongs to $\Lambda(\mathbf{F})$ if and only if n is even, i.e., the graph $\mathcal{C}_n^1$ is bipartite.

On the other hand, when the state-space model 2 is evaluated, the spectrum of the state matrix $\mathbf{F}'(\eta)$ derives from (3.32) through a linear convex transformation and results to be

$$\Lambda(\mathbf{F}'(\eta)) = \left\{ \lambda_i' \in \mathbb{R} \mid \lambda_i' = \eta + (1-\eta)\cos\left(\frac{2\pi}{n} i\right),\ i = 0 \dots n-1 \right\}. \tag{3.33}$$

As a consequence, the largest negative eigenvalue in modulus is $\lambda_{n/2}'$ when number n of cameras is even and $\lambda_{(n-1)/2}'$ when it is odd, whereas the largest positive eigenvalue (excluding $\lambda_0' = 1$) is always λ_1'.

It can be shown that when the system cardinality is even, the second largest eigenvalue in modulus is $\lambda'_{n/2}$ by imposing

$$\eta \leq \frac{\left(1-\cos\left(\frac{\pi}{n}\right)\right)\left(1+\cos\left(\frac{\pi}{n}\right)\right)}{1+\left[\left(1-\cos\left(\frac{\pi}{n}\right)\right)\left(1+\cos\left(\frac{\pi}{n}\right)\right)\right]} = \eta_e^{cr}. \tag{3.34}$$

If the cardinality of the camera network is even, indeed, the largest negative eigenvalue is $\lambda'_{n/2} = -1 + 2\eta$. As a consequence, this rules the convergence of the rotation angle estimates toward the equilibrium values only if it holds that $|\lambda'_{n/2}| \geq |\lambda'_1|$, where $\lambda'_1 = \eta + (1-\eta)\cos\left(\frac{2\pi}{n}\right)$. This means

$$1 - 2\eta \geq \eta + (1-\eta)\cos\left(\frac{2\pi}{n}\right) \quad \text{or equivalently,} \tag{3.35}$$

$$\eta \leq \frac{1-\cos\left(\frac{2\pi}{n}\right)}{3-\cos\left(\frac{2\pi}{n}\right)} = \frac{\left(1-\cos\left(\frac{\pi}{n}\right)\right)\left(1+\cos\left(\frac{\pi}{n}\right)\right)}{1+\left[\left(1-\cos\left(\frac{\pi}{n}\right)\right)\left(1+\cos\left(\frac{\pi}{n}\right)\right)\right]}. \tag{3.36}$$

A similar reasoning can be carried out when the network is constituted by an odd number of cameras. In such a case, the negative largest eigenvalue (in modulus) is $\lambda'_{(n-1)/2} = -\left(\eta + (1-\eta)\cos\left(\frac{2\pi}{n}\frac{n-1}{2}\right)\right) = \eta - (1-\eta)\cos\left(\frac{\pi}{n}\right)$. Therefore, imposing that $|\lambda'_{(n-1)/2}| \geq |\lambda'_1|$, it results

$$-\eta + (1-\eta)\cos\left(\frac{\pi}{n}\right) \geq \eta + (1-\eta)\cos\left(\frac{2\pi}{n}\right) \quad \text{or equivalently,} \tag{3.37}$$

$$\eta \leq \frac{-\left[\cos\left(\frac{2\pi}{n}\right)-\cos\left(\frac{\pi}{n}\right)\right]}{2-\left[\cos\left(\frac{2\pi}{n}\right)-\cos\left(\frac{\pi}{n}\right)\right]} = \frac{\left(1-\cos\left(\frac{\pi}{n}\right)\right)\left(\frac{1}{2}+\cos\left(\frac{\pi}{n}\right)\right)}{1+\left[\left(1-\cos\left(\frac{\pi}{n}\right)\right)\left(\frac{1}{2}+\cos\left(\frac{\pi}{n}\right)\right)\right]}. \tag{3.38}$$

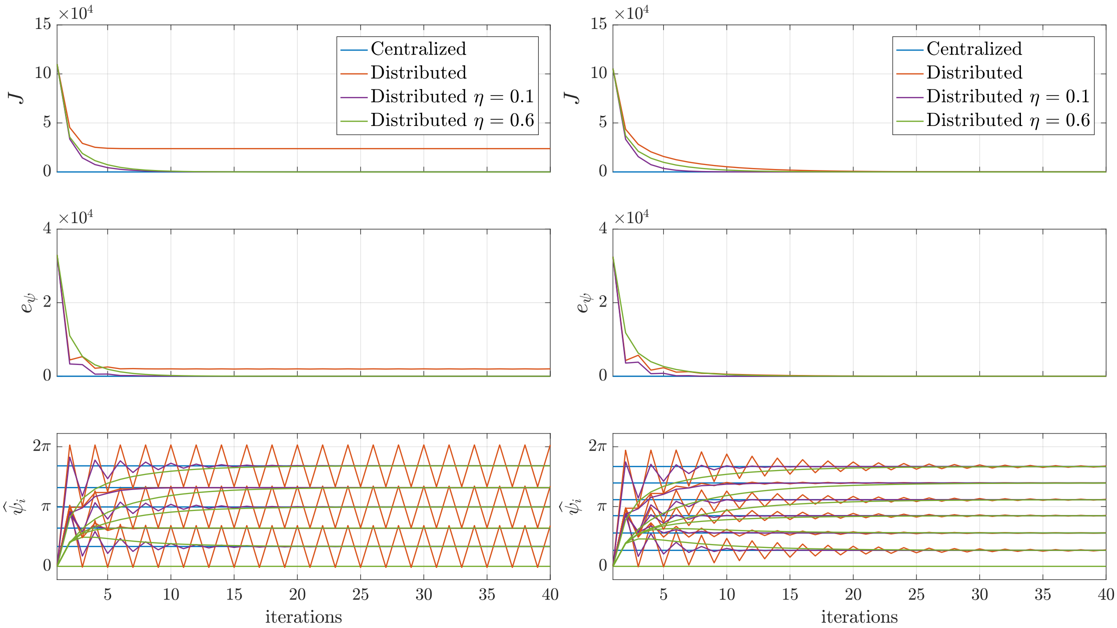
Summarizing, when the network is made up of an odd number of nodes (i.e., devices), then $\lambda'_{(n-1)/2}$ is the second largest eigenvalue if

$$\eta \leq \frac{\left(1-\cos\left(\frac{\pi}{n}\right)\right)\left(\frac{1}{2}+\cos\left(\frac{\pi}{n}\right)\right)}{1+\left[\left(1-\cos\left(\frac{\pi}{n}\right)\right)\left(\frac{1}{2}+\cos\left(\frac{\pi}{n}\right)\right)\right]} = \eta_o^{cr} \tag{3.39}$$

In practice, η parameter allows to control the convergence performance to the asymptotic solution in terms of speed and damped oscillations. These observations can be summarized into the following lemma.

Lemma 3.5.4. *Given a circulant graph $\mathcal{C}_n^1$ that models a camera network of n devices and the state-system model* (3.24) *that describes the dynamics of the pan-only calibration, the unique equilibrium configuration is reached through damped oscillatory convergence behavior if parameter η is below a critical value. Moreover, in this situation, the convergence speed is computed by eigenvalue analysis. Formally*

- *if n is odd and η satisfies* (3.39) *then the dominant mode is $m(k) = \left(\lambda'_{(n-1)/2}\right)^k \;\; k \in \mathbb{Z}$;*
- *if n is even and η satisfies* (3.34) *then the dominant mode is $m(k) = \left(\lambda'_{n/2}\right)^k \;\; k \in \mathbb{Z}$.*



(a) $n = 6$ **(b)** $n = 7$

Figure 3.10. Models (3.20) and (3.24): convergence analysis for a network represented by (a) $\mathcal{C}_6^1$ graph and by (b) $\mathcal{C}_7^1$ graph.

3.5.4 Numerical Results

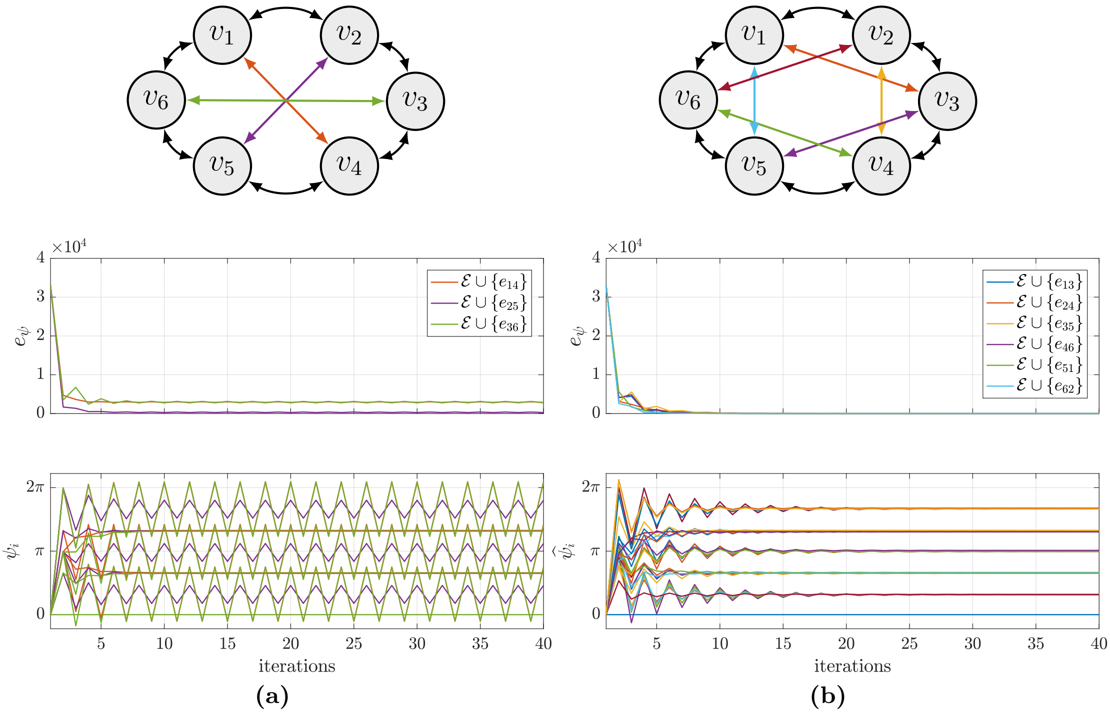
To compare the two state-system update models (3.20) and (3.24), an iterative estimation of n rotation angles is performed considering a circulant graph-represented network, starting from zero initial conditions, i.e., $\widehat{\psi}_i(0) = 0, i = 1 \dots n$, and using the centralized solution (3.18) as ground truth. Moreover, all the relative pan angles are assumed to be corrupted by an additive noise uniformly distributed in the range $[-\frac{\pi}{36}, +\frac{\pi}{36}]$ and the estimation procedure stops after 40 iterations.

To evaluate the convergence properties, a performance index is introduced, namely the *mean squared error*. At each iteration t, this is defined as the squared distance between the current estimated angles $\{\widehat{\psi}_i(t)\}$ and the real values $\{\psi_i\}$ given by the unique centralized solution, i.e.,

$$e_\psi(t) = \frac{1}{n}\sum_{i=1}^{n}(\widehat{\psi}_i - \psi_i)^2. \tag{3.40}$$

Figure 3.10 reports the estimation results in terms of cost function J, mean squared error e_ψ and trend of the estimated angles normalized with respect to the first angle ($\psi_1 = \widehat{\psi}_1 = 0$) for a network consisting of an even ($n = 6$) and an odd ($n = 7$) number of devices, respectively. In both cases, the iterative estimation is executed considering the model (3.20) (orange line) and the model (3.24) for two different values of the parameter η that satisfy (purple line) or not (green line) conditions (3.34)-(3.39).

By applying the system update law (3.20) to the 6-cameras system, it can be observed that the cost function J does not converge to zero and the trend of the error e_ψ regularly oscillates around a non-zero value as the estimates of some angles do not reach the



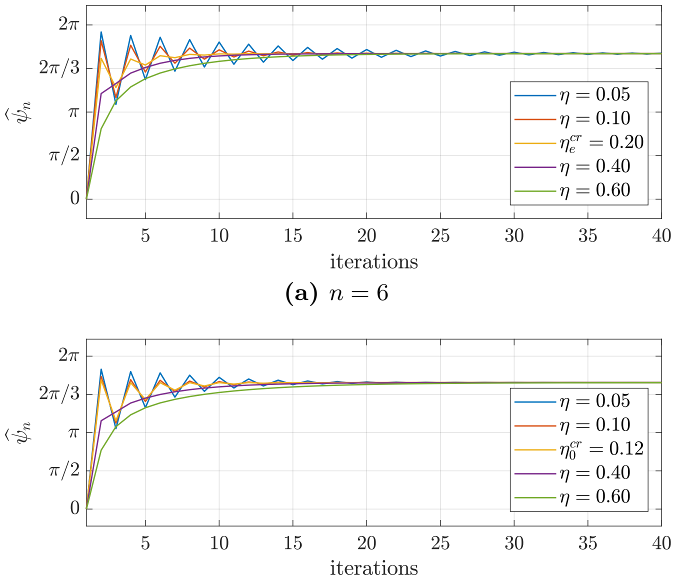
(a) **(b)**

Figure 3.11. Model (3.20): effects on the convergence performance due to a change in the topology of an initially $\mathcal{C}_6^1$ graph. The insertion of an additional link can still produces bipartite graphs (a) or can give rise to a non-bipartite graph (b).

equilibrium configuration and synchronously oscillate around their true values (orange lines in the plots of Figure 3.10a). This behavior is caused by the presence of the eigenvalue $\lambda = -1$ in the spectrum of the state matrix $\mathbf{F}$ governing the dynamics of the network, which is described by the bipartite graph $\mathcal{C}_6^1$. Conversely, when there is at least one cycle with an odd number of nodes as in a $\mathcal{C}_7^1$ graph, the convergence is assured for the angles estimates as shown in the plots of Figure 3.10b (orange line).

On the other side, adopting the model (3.24), the cost function (3.13) and the performance index (3.40) decrease to zero for both circulant networks with $n = 6$ and $n = 7$. Note that the transient period of e_ψ in correspondence to $\eta = 0.1 < \eta_e^{cr}, \eta_o^{cr}$ (purple line): such a parameter setting ensures the second largest eigenvalue in modulus is negative motivating the damped oscillatory convergent behavior of the angles.

Interestingly, when the model (3.20) is considered in the $\mathcal{C}_6^1$ case, an accurate choice of network topological changes can be employed to solve the convergence issue. As a matter of fact, in such a scenario, connections can be added to the network configuration with the purpose of generating at least one odd-length cycle. Figure 3.11 shows the effects of all possible additional links in the 6-cameras circulant network: while any of the additional connections in Figure 3.11a does not help to solve the problem since all the cycles still consist of an even number of nodes, any of the topologies obtained in Figure 3.11b yields the generation of at least one cycle with odd cardinality. As expected from the theoretical discussion in Section 3.5.2, this entails the elimination of the eigenvalue $\lambda = -1$ from the



(a) $n = 6$

(b) $n = 7$

Figure 3.12. Model (3.24): convergence analysis of a single pan angle in $\mathcal{C}_6^1$ and $\mathcal{C}_7^1$ network scaling η parameter around its critical value.

spectrum of matrix state $\mathbf{F}$ assuring both the convergence of the e_ψ index to zero and of the angles towards their asymptotic true values.

As a final issue, the application of Lemma 3.5.4 is proposed to both the $\mathcal{C}_6^1$ and $\mathcal{C}_7^1$ cases, reported in Figures 3.12a-3.12b respectively. Here, the role of the η parameter is shown with respect to the convergence of a single pan angle estimate towards its true value. According to Lemma 3.5.4, oscillatory damped behavior is obtained for values of the parameter lower than the critical one (blue and orange lines), and in particular the settling time inversely depends on η. On the contrary, the convergence speed is directly proportional to the control parameter when the conditions (3.34)-(3.39) are not fulfilled.

Remark 3.5.5. The assumption of circulant topologies in camera networks is in practice well justified since in the majority of real scenarios the field of views of neighboring devices are shared (e.g., outdoor perimeter surveillance, indoor monitoring of structured environments). Similarly, when it comes to model wider FoVs or visibility ranges, it is more realistic to add edges that link to the second nearest neighbors (for example) than to nodes that are far away in the topology structure, hence a situation like that of Figure 3.11b is favoured with respect to the one depicted in Figure 3.11a.

3.6 Results Summary

In this chapter, the VSNs are introduced as multi-device architectures wherein each element is a smart camera with fixed position and adjustable orientation. For this reason, the attitude estimation problem for a n-cameras network is addressed. This consists in the reconstruction of the absolute orientation for each device in the system, given the availability of noisy relative measurements between cameras having overlapping FoVs.

In the first part, cameras acting in the 3D space are considered. Several initialization methods are proposed for the minimization-based iterative procedure illustrated in Tron and Vidal (2009, 2014), which can also be used per se as a non-iterative solution, in particular when the noise level is high. The findings obtained through the numerical simulations with ideal Gaussian noise distribution on the measurements are consistent with those observed in a real implementation, where the relative rotation measurements are corrupted by realistic noise. The results achieved by employing these algorithms confirm that the a priori information on the relative rotations, which is commonly available as the input measurements, can be profitably exploited to obtain good estimates for the cameras attitude. More in detail, the proposed averaged methods, aSST and aMP, outperform the corresponding non-averaged versions, providing smaller mean errors at the cost of an additional burden in the algorithm complexity.

In the second part of the chapter, the analysis is restricted to the 2D scenario where each rotation matrix $\mathbf{R} \in SO(2)$ is univocally determined by the pan angle ψ and an interesting connection emerges between iterative convex minimization procedures and discrete time state-space models characterized by stochastic state matrices. Rewriting the solution proposed by Tron and Vidal (2009, 2014) for the bi-dimensional case, an iterative approach is derived relying on two different state-space models for the update of the estimates, whose convergence properties rest upon the eigenvalue analysis of a particular class of stochastic matrices. More specifically, the convergence behavior of the first iterative scheme (3.20) can be partially governed by introducing limited modification to the camera network connections, while the second model (3.24) natively provides a tuning parameter that allows to control the convergence properties of the solutions towards the optimal estimation values. The numerical simulation results support the theoretical considerations about the convergence performance.

4

Perimeter Patrolling Task for Calibrated VSNs

The surveillance task deals with the real-time monitoring of static and dynamic elements in a specific area of interest in order to understand and predict their future actions and interactions. The employment of smart camera networks for this purpose has become ubiquitous in many applicative scenarios, from the industrial and military contexts to the public and the commercial environments. It is therefore essential to design intelligent algorithms, capable of self-adapting to a variety of situations with minimal human intervention.

In this chapter the boundary patrolling problem is considered, where a (calibrated) visual sensor network is required to monitor the perimeter so as to detect anomalies and track possible intrusions. To model real-world conditions, each camera of the considered system is assumed to be characterized by both physical mobility range and limited speed.

The contents of this chapter are available in
Belgioioso G., Cenedese A., and Michieletto G. Distributed partitioning strategies with visual optimization for camera network perimeter patrolling. *IEEE 55th Conference on Decision and Control (CDC)*, pages 5912–5917, 2016.

4.1 Perimeter Patrolling Task

Patrolling refers to the act of walking or travelling through an area, at regular intervals, with the aim of monitoring and surveillance (Chevaleyre (2004)). In recent years, this task has been applied to artificial systems such as robotic and camera networks being interpreted as a multi-agent cooperative issue to be accomplished in a robust and efficient manner.

4.1.1 Literature Overview

In the last decades, different distributed strategies solving the patrolling problem have been proposed in the literature, which differ in the nature of the employed devices and the specific constraints imposed to the task (Kingston, Beard, and Holt (2008); Pasqualetti, Zanella, Peters, Spindler, Carli, and Bullo (2014); Azad, Casari, and Zorzi (2012)). However, the attention has been mainly focused on the networked camera system. Indeed, several automated surveillance systems based on PTZ cameras have already been introduced in public, industrial and home contexts with the purpose of patrolling the perimeter or a portion of a structure, in order to detect and track an intruder with limited or no human involvement (see e.g., Biswas, Guha, Mukerjee, and Venkatesh (2006); Baseggio, Cenedese, Merlo, Pozzi, and Schenato (2010); Taj and Cavallaro (2011)).

In the most studied robotic scenarios, the autonomous agents are able to move within an area or along a boundary of interest and aim at cooperatively partition and explore the space according to their dynamics while maximizing some optimality measure (Pavone, Arsie, Frazzoli, and Bullo (2011); Acevedo, Arrue, Maza, and Ollero (2013); Di Fava, Satler, and Tripicchio (2015), to cite a few). Differently, in the camera system framework, the agents are constrained not only by their actuation dynamics but also by their installation positions and the problem is again to compute the optimum portion of the scene to be monitored through the elaboration of locally exchanged information. Interestingly, in both the scenarios, the solution is usually sought by recasting the optimal *patrolling* problem in terms of an optimal *partitioning* problem as also discussed in Czyzowicz, Gasieniec, Kosowski, and Kranakis (2011). Specifically, perimeter (rather than area) patrolling task is a one dimensional problem dealing with the real-time monitoring of persistent and transient objects constrained along the boundary of a certain portion to be protected which is is often treated as a finite length segment. Hence, the solution to the desired task may be computed by solving a segment partition optimization problem.

Without pretending to be exhaustive and focusing on this approach, from the literature a good strategy results be the minimization of the time interval between successive inspections of the same point on the perimeter. This is the the rationale behind the solutions proposed in Baseggio et al. (2010); Carli, Cenedese, and Schenato (2011); Alberton, Carli, Cenedese, and Schenato (2012); Bof, Carli, Cenedese, and Schenato (2017), where some algorithms for the optimal segment partitioning are presented with a particular attention to the adopted communication protocols.

4.1.2 Problem Formalization

In the smart multi-camera systems framework described in Section 3.1, the solution proposed in this chapter for the perimeter patrolling problem follows the same steps paved in Carli et al. (2011) and the aforementioned similar works, but, at the same time, considers other fundamental aspects. First of all, the original perimeter patrolling problem is here addressed without recasting it into a segment partitioning problem. In other words, differently from the previously cited literature, the perimeter is actually described as a Jordan curve and not as a segment. Secondly, the optimality criterion is redefined in a more consistent and comprehensive fashion: in addition to the minimization of the time interval between two successive inspections of the same point, also a penalty function related to the quality of vision of the cameras along the perimeter is considered. Through this formalization an optimal solution can be analytically obtained.

Let consider a n-nodes visual sensor network. According to the discussion carried out at beginning of Section 3.5, each camera c_i (with $i = 1 \dots n$) in the system can be modeled as a rigid body with partial motion capability acting in 2D space (Figure 3.8). Each device can indeed vary its orientation, univocally determined by the pan angle, while its position is fixed. The variation of the pan angle is thus the unique DoF of the camera that has point FoV. In addition, it is assumed that pan movements do not alter the view perspective mening that the device is characterized by fixed coverage range.

The perimeter patrolling task requires to monitor a perimeter $\mathscr{L}$ that is assumed to be a closed line (a Jordan curve) of length $l \in \mathbb{R}^+$, wherein each point $s \in \mathscr{L}$ is uniquely identified by the distance, along the perimeter in a counterclockwise direction, which separates it from an arbitrarily chosen starting point, denoted as $s_0 = 0$.

On the devices side, let introduce the quantities represented in Figure 4.1 that characterize each visual sensor node in the patrolling network.

- $D_i = [\underline{d}_i,\ \overline{d}_i] \subset \mathscr{L}$ with $\underline{d}_i < \overline{d}_i$ is the *patrolling range* (PR) of the i-th camera, i.e., its total coverage range due to the scenario topology and its physical features;
- $A_i = [\ell_i,\ r_i]$ with $\ell_i \leq r_i$ is the *patrolling area* (PA) of the i-th camera, i.e., the area actively patrolled by the device which, differently from D_i, can be updated over the time (namely $A_i = A_i(t)$) under the physical constraint $A_i \subseteq D_i$;
- $v_i(t) \in \mathbb{R}^+$ is the i-th camera speed during the pan movement at the t-th instant[1] and $\overline{v}_i \in \mathbb{R}^+$ is its *maximum speed*, so that $|v_i(t)| \leq \overline{v}_i \ \forall\ t$.

To formalize the patrolling issue within an optimization framework, it is necessary to establish a suitable optimality criterion. Following Carli et al. (2011) and similar works, an appropriate choice is the minimization of the maximum time interval (*lag time*) between two consecutive visits of the same point of the perimeter, namely the minimization of the cost function $T_{lag} \colon \mathscr{L} \to \mathbb{R}_0^+$ such that $T_{lag}(s)$ is equal to zero if location $s \in \mathscr{L}$ is currently monitored by any camera.

[1] Note that in this chapter, conversely to the previous one, the notation v_i is not used to indicate the i-th camera in the network. This is referred as c_i as stated before.

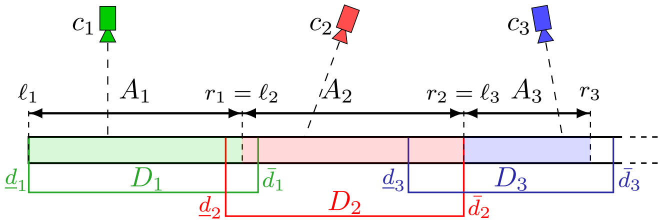


Figure 4.1. Patrolling ranges $\{D_i\}$ and the patrolling areas $\{A_i\}$ representation for the first three cameras of the perimeter surveillance system.

It has been proved that the problem of globally minimizing T_{lag} turns into the problem of designing the optimal patrolling areas $\{A_i\}_{i=1}^n$ for the multi-agent camera network. Specifically, this fact is justified by the Proposition II.1 in Carli et al. (2011), which states that the smallest lag time for the patrolling area $A_i = [\ell_i,\ r_i]$ of the i-th camera is attained assuming that the device moves at its maximum speed $\overline{v}_i$, forward and backward along A_i with a periodic motion of period $T^*_{lag}(A_i) = 2|A_i|/\overline{v}_i$. As a consequence, the patrolling problem can be formulated as a partitioning problem, namely

$$
\begin{aligned}
(\mathsf{P}_1):\quad & \min_{\{A_i\}_{i=1}^n} \ \max_i\{T^*_{lag}(A_i)\} && (4.1)\\
& \text{s.t.}\ \ A_i \subseteq D_i \quad i = 1 \dots n && (4.2)\\
& \cup_{i=1}^n A_i = \mathscr{L} && (4.3)
\end{aligned}
$$

From the *physical constraints* (4.2), it is possible to derive a necessary condition guaranteeing the feasibility of full coverage. The *covering constraint* (4.3) is indeed ensured when the following assumptions are in place:

$$
\begin{aligned}
& \underline{d}_i \le \underline{d}_{i+1} \le \overline{d}_i \le \overline{d}_{i+1} \quad i = 1 \dots n-1 && (4.4)\\
& \underline{d}_1 = 0 \quad \text{and} \quad \overline{d}_n = l. && (4.5)
\end{aligned}
$$

Interlacing physical constraints (4.4) and *boundary constraints* (4.5) allow to recast problem P_1 as a linear program whose optimal solution is unique and is also a solution for P_1, as stated in the next proposition. Nevertheless, the requirements (4.4)-(4.5) also implies that the perimeter $\mathscr{L}$ is modeled as finite segment instead of a closed curve.

Proposition 4.1.1 (Prop. II.3 in Alberton et al. (2012))**.** *Consider the following segment partitioning problem*

$$
\begin{aligned}
(\mathsf{P}_2):\quad & \min_{\{r_i\}_{i=1}^{n-1},\{\ell_i\}_{i=2}^n} \sum_{i=1}^n \frac{(r_i-\ell_i)^2}{\overline{v}_i} && (4.6)\\
& \text{s.t.}\quad \underline{d}_i \le \ell_i, r_i \le \overline{d}_i \quad i = 1 \dots n && (4.7)\\
& \ell_{i+1} \le r_i \quad i = 1 \dots n && (4.8)\\
& \underline{d}_1 = \ell_1 = 0, \quad \overline{d}_n = r_n = l && (4.9)
\end{aligned}
$$

The corresponding set of minimizers $\mathscr{M}_2$ is a singleton and it is such that $\mathscr{M}_2 \subseteq \mathscr{M}_1$, where $\mathscr{M}_1$ is the solution set of problem P_1.

Note that each partition $\{A_i\}_{i=1}^n$ can be equivalently described by the extremes set $\mathcal{X} = \{r_i, \ell_i\}_{i=1}^n$. Using this notation, the cost function (4.6) of problem P_2 is a function of $\mathcal{X}$ and can be written as

$$J_2(\mathcal{X}) = \frac{1}{\overline{v}_1} r_1^2 + \sum_{i=2}^{n-1} \frac{(r_i - \ell_i)^2}{\overline{v}_i} + \frac{1}{\overline{v}_n}(l - \ell_n)^2 \tag{4.10}$$

Function (4.10) is quadratic and positive definite, so also strictly convex. The constraints set $\mathcal{C}_2$ defined by (4.7)-(4.9) is convex, compact and non-empty. These properties ensure that the minimum of (4.10) restricted to $\mathcal{C}_2$ exists and is unique. Moreover, such a minimizer $\mathcal{X}_2^* \in \mathscr{M}_2$ is also a solution for P_1 (see Alberton et al. (2012) for a formal proof): considering P_2 instead of P_1 is however more convenient as it can be solved in a distributed manner through iterative algorithms converging to a unique point.

4.2 From Segment To Perimeter Partitioning

The interlacing and boundary constraints (4.4)-(4.5) identify only the subset of the overall patrolling problems where the left extreme of the patrolling range of the first camera ($\underline{d}_1$) is fixed to 0 and the right extreme of the patrolling range of the last camera ($\overline{d}_n$) is constrained in l. More generally, the set of problems described by P_2 are characterized by the presence of at least one pair of adjacent devices whose patrolling ranges share only the border. In these frameworks, the perimeter to be monitored is considered as a finite segment, hence the problem to solve is the *optimal segment partitioning*. This section is instead devoted to the extension to the case wherein the patrolling ranges of each pair of adjacent cameras may overlap, so that the perimeter $\mathscr{L}$ might be modeled as a closed curve leading to the determination of the *optimal perimeter partitioning*.

For the sake of simplicity, in the following it is assumed that there exists only one camera whose patrolling range (hence, whose patrolling area) is across the chosen starting point $s_0 = 0$ of the perimeter, assumed to be located after $\overline{d}_n$, so that it still holds $\overline{d}_n = l$. Without loss of generality, let this device be the camera c_1. Under these hypotheses, the patrolling ranges for the devices constituting the surveillance system turn out to be

$$D_1 = [0 \ \overline{d}_1] \cup [\underline{d}_1 \ l], \quad \text{with } \overline{d}_1 < \underline{d}_1, \tag{4.11}$$

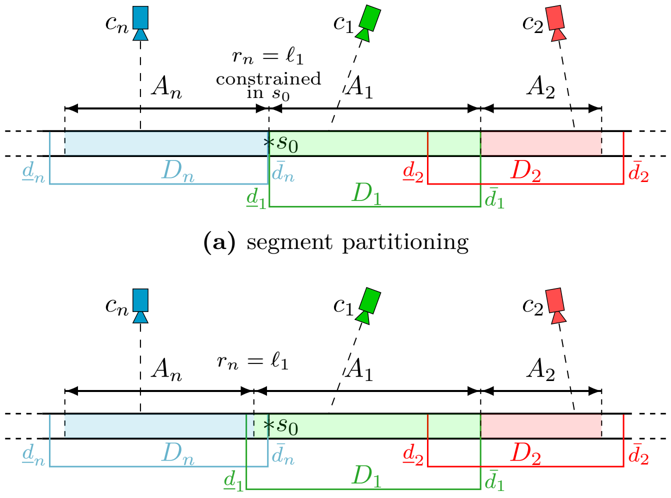
$$D_i = [\underline{d}_i \ \overline{d}_i], \quad \text{with } \underline{d}_i < \overline{d}_i, \quad i = 2 \ldots n. \tag{4.12}$$

As a consequence, the interlacing and boundary constraints (4.4)-(4.5) have to be substituted by the following requirements that include the overlapping condition (4.11) on the patrolling range of the camera c_1 (see Figure 4.2). It must holds that

$$\underline{d}_i \le \underline{d}_{i+1} \le \overline{d}_i \le \overline{d}_{i+1} \quad i = 2 \ldots n-1 \tag{4.13}$$

$$0 \le \underline{d}_2 \le \overline{d}_1 \le \overline{d}_2 \tag{4.14}$$

$$\underline{d}_n \le \underline{d}_1 \le \overline{d}_n = l \tag{4.15}$$



(a) segment partitioning

(b) perimeter partitioning

Figure 4.2. Perimeter patrolling when (a) constraint (4.3) holds and when (b) condition (4.11) is in place.

In this scenario, wherein the perimeter $\mathscr{L}$ is considered as a closed line, the camera network patrolling task can be recast in the following optimization problem

$$(\mathsf{P}_3): \min_{\{\ell_i\}_{i=1}^n \{r_i\}_{i=1}^n} \quad \sum_{i=2}^{n} \frac{[(r_i - \ell_i) \bmod l\,]^2}{\overline{v}_i} \tag{4.16}$$

$$\text{s.t.} \quad \underline{d}_i \leq \ell_i, r_i \leq \bar{d}_i \quad i = 2 \dots n \tag{4.17}$$

$$0 \leq r_1 \leq \bar{d}_1 \tag{4.18}$$

$$\underline{d}_1 \leq \ell_1 \leq \bar{d}_n \tag{4.19}$$

$$\ell_{(i \bmod n)+1} \leq r_i \quad i = 1 \dots n \tag{4.20}$$

Problem P_3 has at least a solution, as proved in the next proposition, which solves also the original problem P_1.

Proposition 4.2.1. *Let $\mathscr{M}_3$ be the set of minimizers of problem P_3. Then $\mathscr{M}_3$ is a non-empty set (containing at least an element) and $\mathscr{M}_3 \subseteq \mathscr{M}_1$, where $\mathscr{M}_1$ is the solution set of problem P_1.*

Proof. The cost function (4.16) of problem P_3 can reformulated as follows

$$J_3(\mathcal{X}) = \frac{(r_1 - \ell_1 + l)^2}{\overline{v}_1} + \sum_{i=2}^{n} \frac{(r_i - \ell_i)^2}{\overline{v}_i}. \tag{4.21}$$

This is quadratic and positive semidefinite since its Hessian is a diagonal block matrix where each block has eigenvalues $\lambda_1 = 0$, $\lambda_2 = 2/\overline{v}_i > 0$. On the other hand, the constraints set $\mathcal{C}_3$ defined by (4.17)-(4.20) is convex, compact and non-empty. These properties ensure the existence of a global minimum for the function (4.21) restricted to $\mathcal{C}_3$, although in general it is not unique.

After these premises, the proof that $\mathscr{M}_3 \subseteq \mathscr{M}_1$ can be conducted following the reasoning carried out in Proposition II.3 in Alberton et al. (2012): it has to be shown that if $\mathscr{X}_3^* = \{r_i^*, \ell_i^*\}_{i=1}^n \in \mathscr{M}_3$, then

$$r_i^* = \ell^*_{(i \bmod n)+1} \quad i = 1 \dots n. \tag{4.22}$$

For $i = 1 \dots n-1$ the proof is exactly the same proposed in Alberton et al. (2012), hence only the case $i = n$ is considered proving that $r_n^* = \ell_1^*$. By contradiction, let assume that $r_n^* > \ell_1^*$, from (4.13)-(4.15) it follows that $\ell_1^* \geq \underline{d}_1 \geq \underline{d}_n$ and $r_n^* \leq \overline{d}_n$. In addition, let introduce the new partition $\mathscr{X}_3'$ such that $\mathscr{X}_3' = \mathscr{X}_3^*$ for all elements except for r_n', which is set to $r_n' = \ell_1^*$. This choice is feasible, i.e., $\mathscr{X}_3' \in \mathcal{C}_3$, since $r_n' \geq \ell_1' = \ell_1^* \geq \underline{d}_n$. This leads to $J_3(\mathscr{X}_3^*) - J_3(\mathscr{X}_3') = \frac{1}{\overline{v}_n}((r_n^* - \ell_n^*)^2 - (\ell_1^* - \ell_n')^2) > 0$, which contradicts the assumption that $\mathscr{X}_3^*$ is a global minimum of J_3 over $\mathcal{C}_3$. □

Proposition 4.2.1 states that at least one optimal solution of problem P_3 exists and is also a solution of the original patrolling problem P_1. Unlike P_2, where the uniqueness of the optimal solution is guaranteed, the general case encoded in P_3 allows infinite configurations for the camera patrolling areas that minimize the lag time T_{lag} and fulfill the condition (4.22).

As a consequence, imposing $r_i = \ell_{(i \bmod n)+1}$ (with $i = 1 \dots n$) problem P_3 can be rewritten in the following alternative form where the variables $x_i = r_i = \ell_{(i \bmod n)+1}$ with $i = 1 \dots n$ are introduced in order to simplify the notation:

$$(\mathsf{P}_3') : \min_{\{x_i\}_{i=1}^n} \frac{(x_1 - x_n + l)^2}{\overline{v}_1} + \sum_{i=2}^n \frac{(x_i - x_{i-1})^2}{\overline{v}_i} \tag{4.23}$$

$$\text{s.t.} \quad \underline{d}_{(i \bmod n)+1} \leq x_i \leq \overline{d}_i \quad i = 1 \dots n \tag{4.24}$$

$$x_i \leq x_{i+1} \quad i = 1 \dots n-1 \tag{4.25}$$

Problems P_3 and P_3' have the same solutions. Moreover, for the unconstrained patrolling ranges case, i.e., when the constraints (4.11)-(4.12) are neglected, the optimal patrolling areas have length proportional to the maximum speed of the related cameras.

Corollary 4.2.2. *Consider problem P_3' without patrolling range constraints, so that $\underline{d}_i = 0$ and $\overline{d}_i = l$ for $i = 1 \dots n$. Let $\mathscr{M}_{3'}$ be the corresponding set of minimizers. Each solution*[2] *$\mathscr{X}_{3'}^* = \{x_i^*\}_{i=1}^n \in \mathscr{M}_{3'}$ admits the following closed form*

$$x_i^* = x_n^* - \frac{\sum_{j=i}^{n-1} \overline{v}_{j+1}}{\sum_{j=1}^n \overline{v}_j} l > 0, \quad i = 1 \dots n-1 \quad \text{and} \quad \frac{\sum_{j=1}^{n-1} \overline{v}_{j+1}}{\sum_{j=1}^n \overline{v}_j} l \leq x_n^* \leq l \tag{4.26}$$

and, consequently, each optimal partition $\{A_i^*\}_{i=1}^n$ satisfies

$$|A_i^*| = \frac{\overline{v}_i}{\sum_{j=1}^n \overline{v}_j} l, \quad i = 1 \dots n. \tag{4.27}$$

[2] With abuse of notation, the extreme set $\mathscr{X} = \{r_i, \ell_i\}_{i=1}^n$ with cardinality $2n$ is hereafter substituted by the extreme set $\mathscr{X} = \{x_i\}_{i=1}^n$ having cardinality n.

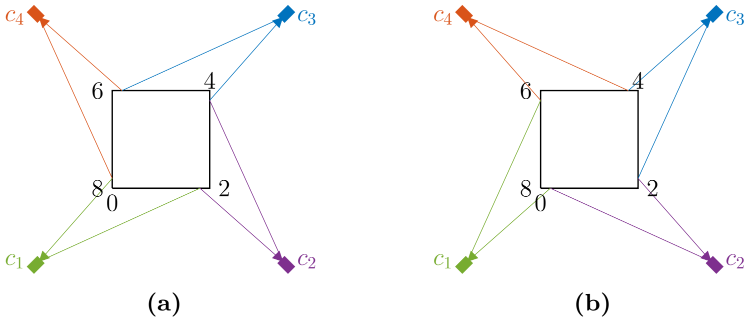


Figure 4.3. Example of multiple solutions for patrolling problem P'_3: a square perimeter of length $l = 8$m patrolled by 4 cameras with equal speeds.

Specifically when the cameras have the same maximum speed ($\overline{v}_i = \overline{v}\ \ \forall i$), then

$$x_i^* = x_n - \frac{n-i}{n} l \quad \textit{and} \quad |A_i^*| = \frac{l}{n}, \quad i = 1 \dots n. \tag{4.28}$$

For the constrained problem P'_3 (as well as P_3) there exist infinite partitions wherein condition (4.27) is satisfied.

Example 3. Let consider the square perimeter $\mathscr{L}$ in Figure 4.3 having length $l = 8$m. To monitor $\mathscr{L}$, a 4-cameras system is used where the devices are placed in correspondence of the square corners. Both the represented partitions in Figure 4.3a and Figure 4.3b are solution of P'_3 and thus are optimal in terms of lag time. Note that the patrolling areas *slide* along the perimeter while preserving their lengths proportional to the cameras speed which is assumed to be equal for all the devices.

◆

The previous example shows that, although iterative distributed algorithms can be implemented to solve P'_3, their convergence is not guaranteed. To overcome this issue and provide a unique solution to the problem, a penalty function based on the visual quality of the cameras on their patrolling areas is introduced in the next section.

4.3 Vision Quality Centering Criterion

From the point of view of the vision quality, it is desirable that each camera patrols the perimeter portion where its view is as much as possible perpendicular. For this reason, the *angle of visual impact* is here introduced.

Given a camera c_i, let $z_i \in D_i$ be the position of its FoV along the perimeter $\mathscr{L}$[3]. The angle of visual impact on z_i is the angle φ_i between the normal to $\mathscr{L}$ at z_i and the line passing through z_i and the device CoM as illustrated in Figure 4.4.

For each i-th camera it is possible to define a function $H_i \colon D_i \to [0,\ \pi/2)$ that associates the absolute value $|\varphi_i|$ of the corresponding angle of visual impact to each point z_i of its

[3] For sake of notation compactness, the time dependency of the camera FoV position is here neglected, using z_i in place of $z_i(t)$.

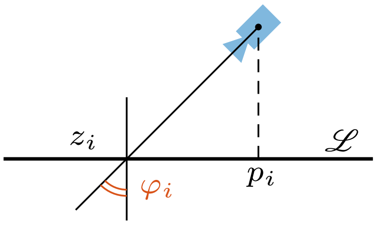


Figure 4.4. Angle of visual impact of the i-th camera: p_i is the position of the camera CoM projected on the perimeter, while z_i is the position of its FoV.

patrolling range. To account for the visual quality over the patrolling area, the penalty index $q \in \mathbb{R}_0^+$ can be determined for the i-th camera as the integral

$$q(A_i) = \int_{A_i} H(z_i)\, dz_i, \tag{4.29}$$

so that the more perpendicular the view of the camera over its patrolling area is, the smaller this factor results.

To determine a unique solution among all the optimal partitions in set $\mathscr{M}_{3'}$, it is suitable to chose the one that provides the best result in terms of penalty index (4.29), namely the solution of the following problem

$$(\mathsf{P}_4): \quad \min_{\{A_i\}_{i=1}^n} \sum_{i=1}^n q(A_i) \tag{4.30}$$

$$\text{s.t.} \quad \{A_i\}_{i=1}^n \in \mathscr{M}_{3'} \tag{4.31}$$

Denoting with $p_i \in \mathscr{L}$ the (fixed) projected position of the i-th camera CoM on the perimeter and assuming that $p_i < p_{i+1}$ with $i = 1 \dots n-1$ and $p_1 \in [0 \ \ \bar{d}_1]$, the following assumptions on the functions H_i are needed to tackle problem P_4: (hp_1) H_i is non-negative and has a unique global minimum at p_i, and (hp_2) H_i is symmetric in a neighborhood of p_i. Under these hypothesis (realistic excluding pathological cases), it occurs that centering the patrolling area of each camera around its p_i ensures the minimum value for $q(A_i)$. Therefore, the penalty index (4.29) can be rewritten in a more convenient manner, namely

$$q(A_i) = \begin{cases} ((p_i - x_n + l) - (x_i - p_i))^2 & i = 1 \\ ((p_i - x_{i-1}) - (x_i - p_i))^2 & i = 2 \dots n \end{cases} \tag{4.32}$$

Note that in this way if the patrolling area A_i is perfectly centered around the position p_i then $(p_i - x_{i-1}) = (x_i - p_i)$ and the penalty factor results to be zero, whereas in all the other cases $q(A_i) \in \mathbb{R}^+$.

Adopting (4.32) instead of (4.29), it is possible to prove that the solution set of P_4 is a singleton; in other words, there is only one partition, among those that minimize the lag time, that provides the best visual quality. Furthermore, the minimization problem (4.30) can be solved analytically to obtain this unique solution in closed form, as formally stated in the next proposition.

Proposition 4.3.1. *Let $\mathscr{M}_4$ be the set of minimizers of problem P_4 under hypothesis (4.13)-(4.15), without patrolling range constraints. If the hypothesis (hp_1), (hp_2) are satisfied for $i = 1 \dots n$, then the solutions set $\mathscr{M}_4$ is a singleton, i.e., $\mathscr{M}_4 = \{\mathcal{X}_4^*\}$.*

Proof. Thanks to (4.32), the cost function of P_4 can be rewritten as a function of the extremes set $\mathcal{X} = \{x_i\}_{i=1}^n$. Specifically, this results to be

$$J_4(\mathcal{X}) = ((p_1 - x_n + l) - (x_1 - p_1))^2 + \sum_{i=2}^{n} ((p_i - x_{i-1}) - (x_i - p_i))^2. \tag{4.33}$$

The function J_4 is quadratic and convex, but not strictly convex. On the other hand, the constraints set $\mathscr{M}_{3'}$ is compact, convex (since P_3 and thus P_3' is a convex problem) and non-empty (from Proposition 4.2.1). Hence there exists at least one optimal solution of problem P_4. Corollary 4.2.2 provides a closed form for a generic optimal partition $\mathcal{X}_{3'}^*$ within $\mathscr{M}_{3'}$, that is a function of x_n. Exploiting (4.26), $J_4(\mathcal{X}_{3'}^*)$ becomes a function of a single unknown variable, x_n, and it is easy to show that it is strictly convex and so it admits a unique global minimum. □

Corollary 4.3.2. *Under the same assumptions of Proposition 4.3.1, the optimal solution $\mathcal{X}_4^* = \{x_i^*\}_{i=1}^n \in \mathscr{M}_4$ of P_4 has the following closed form*

$$\begin{cases} x_i^* = x_n^* - \frac{\sum_{j=i}^{n-1} v_{j+1}}{\sum_{j=1}^{n} v_j} l, & i = 1 \dots n-1, \\ x_n^* = \frac{1}{n} \left(\sum_{j=1}^{N} p_j + \frac{l}{2} + \frac{\sum_{j=1}^{n-1} (\sum_{k=j}^{n-1} v_{k+1})}{\sum_{j=1}^{n} v_j} l \right). \end{cases} \tag{4.34}$$

Specifically, when all the cameras have the same maximum speed, i.e., $\overline{v}_i = \overline{v}\ \ \forall i$, then

$$x_i^* = \frac{1}{n} \sum_{j=1}^{n} p_j - \frac{l}{2} + \frac{l}{n} i, \quad i = 1 \dots n. \tag{4.35}$$

Moreover, if the cameras are set in equidistant manner, i.e., $p_i = \frac{l}{n}(i-1)$ for $i = 1 \dots n$, then

$$x_i^* = \frac{l}{n} \left(i - \frac{1}{2} \right), \quad i = 1 \dots n. \tag{4.36}$$

Once the optimal partition in terms of both lag time and visual quality has been identified, it is possible to define a new distributed optimization problem P_5 resting upon the minimization of the regularized sum of J_3' in (4.23) and J_4 in (4.33). Formally, the following cost function is considered

$$J_5 = J_3' + \frac{1}{\eta} J_4, \tag{4.37}$$

where the constant parameter $\eta \in \mathbb{R}$ allows to weight the penalty term, namely to regularize the optimization problem.

Next proposition shows that P_5 has a unique optimal solution $\mathcal{X}_5^*(\eta)$ that depends on the choice of parameter η and converges to $\mathcal{X}_4^*$ when η increases to infinite and the cameras maximum speeds are equal.

Proposition 4.3.3. *Consider problem* P_5 *under hypotheses* (4.13)-(4.15)*, without patrolling range constraints. Assume that* (hp_1)*,* (hp_2) *are satisfied for* $i = 1 \dots n$*. Let* $\mathscr{M}_5$ *be the corresponding set of minimizers. Then*

a) $\mathscr{M}_5$ *is a singleton, i.e.,* $\mathscr{M}_5 = \{\mathcal{X}_5^*(\eta)\}$*;*

b) when $\overline{v}_i = \overline{v}\ \ \forall i$*, it holds that* $\lim_{\eta\to\infty} \mathcal{X}_5^*(\eta) = \mathcal{X}_4^*$*.*

Proof. a) Cost function (4.37) is quadratic and positive definite: its Hessian is strictly diagonally dominant with real positive diagonal entries. The considered constraints set is non-empty, compact and convex. Hence P_5 is a convex optimization problem and admits a unique solution, $\mathcal{X}_5^*(\eta)$, whose value depends on the parameter η.

b) Since J_5 in (4.37) is quadratic, the explicit solution of P_5 in the unconstrained case can be found by solving the linear system $\mathbf{H}_{J_5}\mathbf{x}^* - \mathbf{a} = 0$; where $\mathbf{H}_{J_5} \in \mathbb{R}^{n\times n}$ is the Hessian of J_5 (namely $\mathbf{H}_{J_5} = \nabla^2 J_5(\mathbf{x})$ where $\mathbf{x} = [x_1\ \dots\ x_n]^\top \in \mathbb{R}^n$ is the vector of the extremes in $\mathcal{X}$), $\mathbf{a} \in \mathbb{R}^n$ is a vector of known coefficients and $\mathbf{x}^* \in \mathbb{R}^n$ is the vector stacking the optimal extremes. It can be derived that the matrix $\mathbf{H}_{J_5}$ is non-singular, therefore the optimal partition results to be $\mathbf{x}^* = \mathbf{H}_{J_5}^{-1}\mathbf{a}$.

In case of equal cameras maximum speeds ($\overline{v}_i = \overline{v}\ \ \forall i$), $\mathbf{H}_{J_5}$ becomes a *tridiagonal symmetric circulant matrix*, identified by the row vector $[a\ b\ 0\ \dots\ 0\ b]^\top \in \mathbb{R}^n$, with $a = 2(k+1)$, $b = (1-k)$ and $k = \eta/\overline{v} > 0$. Its inverse is again a symmetric circulant matrix whose form is given in Corollary 3.7 in Carmona, Encinas, Gago, Jiménez, and Mitjana (2015). This is associated to the parameters

$$g_j(a,b,0) = (-1)^j \frac{\left(U_{j-2}\left(\frac{a}{2b}\right) + (-1)^n U_{n-j}\left(\frac{a}{2b}\right)\right)}{2b(1-(-1)^n)T_n\left(\frac{a}{2b}\right)} \qquad j = 1 \dots n, \tag{4.38}$$

where $T_j(\cdot)$ and $U_j(\cdot)$ are respectively the *first and second kind Chebyshev polynomials.* Therefore the components of solution vector $\mathbf{x}^*$ can be expressed in explicit form as

$$\begin{aligned} x_i^*(k) = \big(\,&(k+1)g_{(n-i+1)} - (k-1)g_{((1-i)\mathrm{mod}n+1)}\big)L \\ &+ \sum_{j=1}^{n} (p_j p_{(j\mathrm{mod}n+1)})g_{((j-i)\mathrm{mod}n+1)}. \end{aligned} \tag{4.39}$$

Using this last expression it can be verified that $\lim_{k\to\infty} x_i^*(k) = x_i^{**}$, where x_i^{**} is defined in (4.35), which proves the thesis. □

4.4 Symmetric-Gossip Partitioning Algorithm

This section is devoted to the presentation of an iterative strategy, named *Symmetric-Gossip Partitioning Algorithm* (s-PAC), that can be run by a n-nodes camera system in a distributed way in order to partition a given perimeter taking into account the lag time minimization jointly with the optimization of the visual quality criterion.

It is assumed that each camera c_i can communicate with cameras $c_{(i-1)\mathrm{mod}n}$ and $c_{(i+1)\mathrm{mod}n}$ (namely the considered network topology is represented by a circulant graph having a single cycle, i.e., $\mathcal{C}_n^1$), and that the patrolling areas can be updated using only local information exploiting a symmetric gossip communication protocol (introduced in Section 1.2). Furthermore, it is assumed that each camera[4] c_i can store in memory the extremes, ℓ_i and r_i, of its patrolling area, jointly with the maximum speed and the center position of its neighbors, i.e., $\overline{v}_{i+1}$, $\overline{v}_{i-1}$ and p_{i+1}, p_{i-1}, which are supposed to be known.

Finally, the fulfillment of the physical and the interlacing constraints is required at each iteration of the algorithm to ensure the convergence towards the optimal solution.

4.4.1 Algorithm Structure

The s-PAC structure consists of the following three phases: the first one is executed only once at the beginning of the procedure, while the last two are repeated at each iteration. The maximum number of executing iterations is a priori fixed and constitutes one of the tunable parameters of the algorithm.

Initialization Phase

At $t = 0$ (initial iteration), the patrolling areas are initialized so that they correspond to the patrolling ranges, namely $\{A_i(0)\}_{i=1}^n = \{D_i(0)\}_{i=1}^n$. As a consequence, it follows that initial partition satisfies

1. the physical constraints (4.2), namely

$$A_i(0) \subseteq D_i(0) \qquad i = 1 \dots n; \tag{4.40}$$

2. the interlacing logical constraints given by

$$\ell_i(0) \leq \ell_{i+1}(0) \leq r_i(0) \leq r_{i+1}(0) \tag{4.41}$$

$$0 \leq \ell_2(0) \leq r_1(0) \leq r_2(0) \tag{4.42}$$

$$\ell_n(0) \leq \ell_1(0) \leq r_n(0) \leq l \tag{4.43}$$

Observe that because of (4.40) and (4.41)-(4.43) the initial set $\{A_i(0)\}_{i=1}^n$ satisfies also the covering constraint (4.3).

Transmission Phase

During each t-th iteration, only a pair of neighboring cameras, c_i and c_{i+1}, communicates. Specifically, c_i sends to c_{i+1} the values of its extremes $\ell_i(t)$ and $r_i(t)$ and viceversa.

Extremes Update Phase

When the transmission phase is concluded, each camera c_h, $h \neq i, i+1$ keeps its extremes unchanged ($\ell_h(t+1) = \ell_h(t)$ and $r_h(t+1) = r_h(t)$). On the contrary, cameras c_i and c_{i+1} update their extremes r_i and ℓ_{i+1} through a two steps procedure.

[4] Hereafter all the subscript indexes are considered modulo n.

(i) They compute the point x^* as

$$x^* = x_T^* - \frac{1}{\eta}(x_T^* - x_V^*) \left(\frac{\overline{v}_i + \overline{v}_{i+1}}{2\overline{v}_i\overline{v}_{i+1}} + \frac{1}{\eta} \right)^{-1} \tag{4.44}$$

The first term x_T^* is responsible for the optimization of the lag time function and satisfies the *equal traveling time criterion* introduced in Alberton et al. (2012) according to which 'the time required to camera c_i to travel at the speed $\overline{v}_i$ from x^* to $\ell_i(t)$ is equal to the time required by the camera c_{i+1} to travel at speed $\overline{v}_{i+1}$ from x^* to $r_{i+1}(t)$'. Formally, x_T^* fulfills the condition $\frac{x_T^* - \ell_i(t)}{\overline{v}_i} = \frac{r_{i+1}(t) - x_T^*}{\overline{v}_{i+1}}$ which yields

$$x_T^* = \frac{\ell_i(t)\overline{v}_{i+1} + r_{i+1}(t)\overline{v}_i}{\overline{v}_i\overline{v}_{i+1}}. \tag{4.45}$$

The second term in (4.44), whose relevance can be weighted by parameter η, is the correction term related to the introduction of the vision quality centering criterion. In particular, the term that takes into account the centering criterion alone is

$$x_V^* = (p_i + p_{i+1}) - \frac{\ell_i(t) + r_{i+1}(t)}{2} \tag{4.46}$$

which is derived by imposing

$$|\ell_i(t) - p_i| = |p_i - r_i(t)| \tag{4.47}$$

$$|\ell_{i+1}(t) - p_{i+1}| = |p_{i+1} - r_{i+1}(t)| \tag{4.48}$$

(ii) Both the cameras c_i and c_{i+1} check if the intervals $[\ell_i(t) \;\; x^*] \subseteq D_i$ and $[\,x^* \;\; r_{i+1}(t)] \subseteq D_{i+1}$ satisfy the physical constraints and they update their extremes r_i and ℓ_{i+1}, respectively, by setting

$$r_i(t+1) = \ell_{i+1}(t+1) = \begin{cases} x^* & x^* \in [\,\underline{d}_{i+1} \;\; \overline{d}_i] \\ \overline{d}_i & x^* > \overline{d}_i \\ \underline{d}_{i+1} & x^* < \underline{d}_{i+1} \end{cases} \tag{4.49}$$

4.4.2 Discussion

If $\overline{v}_i = \overline{v}\ \forall i$ (equal maximum speed case), introducing $k = \eta/\overline{v}$, the expression (4.44) boils down to

$$x^* = x_T^* - \frac{x_T^* - x_V^*}{k+1}. \tag{4.50}$$

From (4.50) it appears that as the parameter k (and therefore η) increases, the impact of the visual quality centering criterion in the extremes update is lower. Moreover, there exists a trade-off between the relevance of the centering criterion during the algorithm iterations and the proximity of solution $\mathcal{X}_5^*(\eta)$ to $\mathcal{X}_4^*$.

These observations allow to introduce the following proposition that states an upper bound for η based on the accuracy required for the system.

Proposition 4.4.1. *Assume that $\overline{v}_i = \overline{v}\ \forall i$. Let $\varsigma \in \mathbb{R}$ be the sensitivity of the cameras with respect to displacements along the perimeter. To ensure at least an accuracy of $\bar{\varsigma} \in \mathbb{R}$ in the centering of the patrolling areas at steady-state (i.e., after the convergence of the algorithm), parameter η has to be set in order to fulfill the following relation*

$$\eta \leq \frac{\bar{\varsigma} - \varsigma}{\varsigma}\,\overline{v} \tag{4.51}$$

Proof. Consider the extremes update step (4.50) during the t-th iteration. In the worst case, the current partition $\mathcal{X} = \{x_i(t))\}_{i=1}^{n}$ optimizes only T_{lag} but not the centering criterion, , i.e., $\mathcal{X} \in \mathscr{M}_3^{\prime}/\mathscr{M}_4$. The only force that drives the extremes to change in (4.50) is $\frac{x_T^* - x_V^*}{k+1}$. To force the extreme update to a different value, it should be

$$\left| \frac{x_T^* - x_V^*}{k+1} \right| \geq \varsigma\,,$$

otherwise the variation would be smaller than the camera sensitivity and the extreme would remain at the previous value. To guarantee an accuracy of at least $\bar{\varsigma}$ in the centering of the patrolling area, it must be $\bar{\varsigma}/(k+1) \geq \varsigma$, whence (4.51) follows. □

As a final note, adopting the same reasoning of Theorem IV.1 in Alberton et al. (2012), the convergence properties of the s-PAC are characterized deriving the conditions that ensure deterministic convergence.

Theorem 4.4.2. *Consider the s-PAC. Assume that the initial partition $\{A_i(0)\}_{i=1}^{n}$ satisfies constraints (4.40) and (4.41)-(4.43). Moreover assume that there exists a positive integer number τ such that, for all t, any pair of neighboring cameras (c_i, c_{i+1}), $i \in \{1 \dots n\}$ communicates with each other at least once within the interval $[t, t+\tau)$. Then the evolution $t \to \{A_i(t)\}$ generated by the s-PAC algorithm is such that the physical constraints are satisfied for all t, and the set $\{A_i(t)\}_{i=1}^{n}$ converges asymptotically to the optimal solution $\mathcal{X}_5^*(\eta)$ of P_5.*

Proof. The proof is analogous to that provided for Theorem IV.1 in Alberton et al. (2012), by adapting to the specific case of the newly defined P_5. □

4.4.3 Numerical Simulations

In this section, the results of two numerical tests aimed at validating the correctness and the effectiveness of the s-PAC are provided. In the first test, the algorithm is implemented considering the simplified setup depicted in Figure 4.3, while in the second one a more realistic scenario for the camera perimeter patrolling is evaluated.

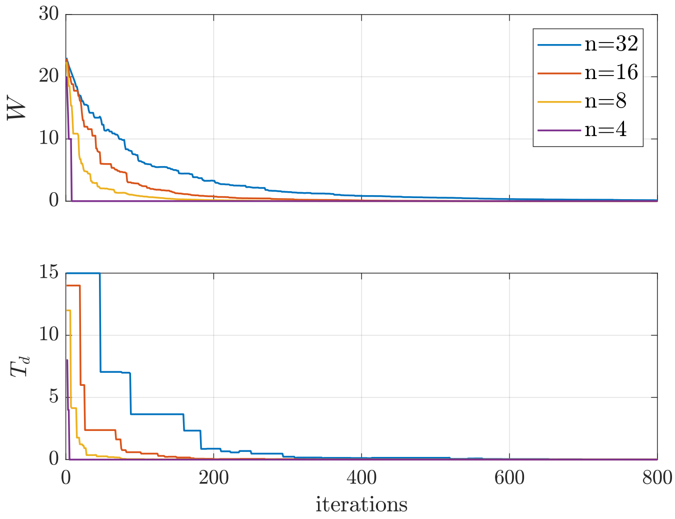


Figure 4.5. Test 1: s-PAC performance on a simulated square setup.

Test 1

A set of n cameras having equal maximum speed $\overline{v} = 1\,\mathrm{m\,s^{-1}}$ is required to patrol a square shape perimeter $\mathscr{L}$, with total length $l = 160$m. It is assumed that at each iteration of the s-PAC the active communication link is randomly chosen and the parameter η is set to 50. The convergence of the procedure to the optimal solution $\mathcal{X}_5^*(\eta)$ is guaranteed by Theorem 4.4.2 and the convergence of $\mathcal{X}_5^*(\eta)$ towards $\mathcal{X}_4^*$ by Proposition 4.3.3.

To analyze the effect of the cardinality n of the camera set on the convergence rate of the s-PAC strategy, the partition algorithm is run setting $n = 8, 16, 32, 64$ and increasing the FoVs correspondingly to ensure the perimeter full coverage. For each run, the following cost function is evaluated

$$W(t) = \sqrt{\frac{1}{2n}\sum_{i=1}^{n}(e_{\ell_i}(t)^2 + e_{r_i}(t)^2)}\,, \tag{4.52}$$

At the t-th iteration, left $e_{\ell_i}(t)$ and right $e_{r_i}(t)$ error terms of camera c_i are computed by

$$e_{\ell_i}(t) = \begin{cases} x_n^*(\eta) - \ell_1(t), & i = 1 \\ x_{i-1}^*(\eta) - \ell_i(t), & i = 2 \dots n \end{cases} \tag{4.53}$$

$$e_{r_i}(t) = x_i^*(\eta) - r_i(t), \quad i = 1 \dots n \tag{4.54}$$

where the optimal partition $\{x_i^*(\eta)\}_{i=1}^n$ is derived using (4.39), while $\ell_i(t)$ and $r_i(t)$ are the i-th left and right extremes during iteration t. Cost function (4.52) is then the standard deviation of the errors between the current extremes and optimal ones, and it conveys information on the distance of the current partition from the unique optimal solution $\mathcal{X}_5^*(\eta)$. The results of the test, represented in Figure 4.5-top, show that the number of iterations needed to reach convergence increases with the cardinality of the patrolling camera set.

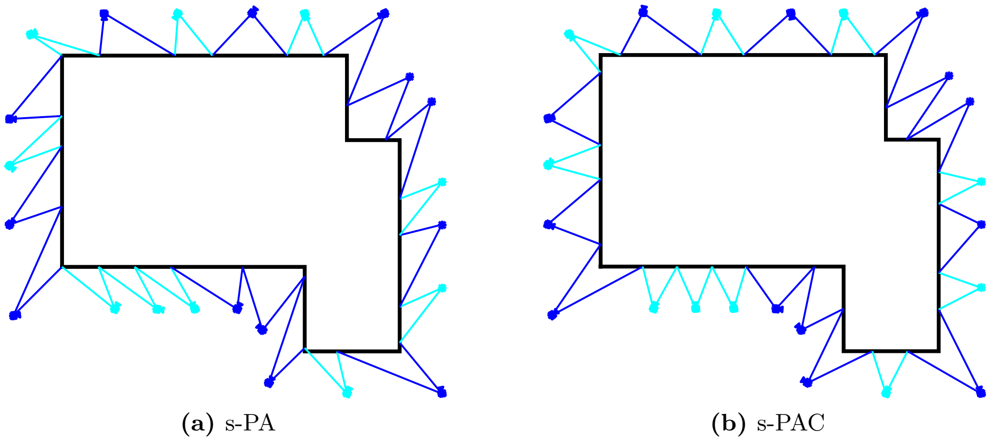


Figure 4.6. Test 2: final partitions obtained using (a) the s-PA procedure and (b) the proposed s-PAC on a realistic scenario with cameras of different speed. Light and dark blue colors indicate respectively slow and fast cameras.

Useful indications on the convergence rate of s-PAC are obtained observing the evolution of the maximum lag time over the patrolling areas, during algorithm execution. In Figure 4.5-bottom the convergence rate of the index $T_d(t) = \max_i\{T_{lag}(A_i(t))\} - T^*_{lag}$ towards zero is represented for different numbers of cameras in the network.

Test 2

This test evaluates a more realistic scenario with a set of $n = 23$ cameras having different maximum speeds, whose goal is to patrol the perimeter $\mathscr{L}$ shown in Figure 4.6, where $l = 60$m. It is assumed again that, at each iteration of the s-PAC, the active communication link is randomly chosen. In this case, while the convergence of s-PAC to $\mathcal{X}^*_5(\eta)$ is ensured by Theorem 4.4.2, the convergence of $\mathcal{X}^*_5(\eta)$ towards $\mathcal{X}^*_4$ is not formally guaranteed since the patrolling devices have different maximum speeds. However simulations results show that this convergence property is attained by increasing the value of parameter η. For this purpose, to distinguish between $\mathcal{X}^*_5(\eta) = \{x^*_i\}^n_{i=1}$ and $\mathcal{X}^*_4 = \{x^{**}_i\}^n_{i=1}$, the following index of proximity is introduced

$$P(\eta) = \frac{1}{\sqrt{n}}\|\mathcal{X}^*_5(\eta) - \mathcal{X}^*_4\|_2 = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x^*_i(\eta) - x^{**}_i)^2} \tag{4.55}$$

The optimal extremes set $\mathcal{X}^*_4$ has been calculated statically using (4.34), whereas $\mathcal{X}^*_5(\eta)$ is the steady-state partition obtained through the simulation of s-PAC on the described scenario. The behavior of index P with respect to η is shown in Figure 4.7: the curve converges exponentially to zero as η increases. Thus, the optimal choice for η falls in the upper bound defined by (4.51), which depends on the cameras sensitivity and accuracy needed for the specific application.

Finally, in order to show the enhancement of visual quality carried by the introduction of the centering criterion, partitions in Figure 4.6 are analyzed: those in Figure 4.6b are obtained performing s-PAC with $\eta = 150$, while those in Figure 4.6a using the s-PA

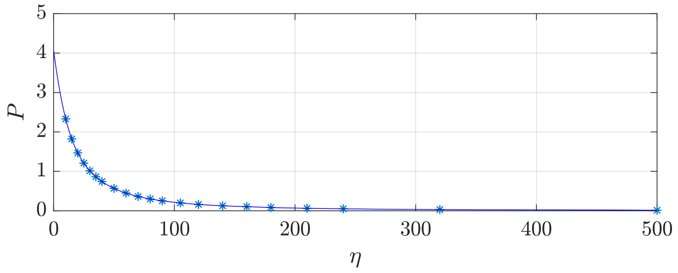


Figure 4.7. Test 2: trend of J index with respect to different value of the regularization parameter $\eta \in [0, 500]$.

strategy introduced in Alberton et al. (2012)[5], which does not take into account the visual quality of the camera during the partitioning procedure. For both the algorithms 500 iterations are executed. The partitions obtained using the two strategies present always larger patrolling areas for faster cameras (indicated with dark blue color in the figures) to optimize T_{lag}, but only the partition identified by the proposed s-PAC show patrolling areas with a good visual quality over the perimeter.

4.5 Results Summary

In this chapter the distributed perimeter patrolling problem is addressed in a multi-camera system context, considering the VSNs introduced in Section 3.5. The proposed solution relaxes the procedure in Baseggio et al. (2010); Carli et al. (2011); Alberton et al. (2012); Bof et al. (2017) from the need to set a starting point on the perimeter implying the emergence of multiple solutions. To solve this issue, a penalty function based on the vision quality of each camera over its patrolling area, is introduced jointly to the minimization of lag time.

This new problem has been mathematically formalized proving the existence of a unique solution for which the analytic form (4.39) has been derived. Finally, a distributed and iterative procedure has been described that allows the cameras network to converge to such optimal solution by exploiting only local communications. Its effectiveness has been proved through numerical simulations, encouraging current effort towards real-world experiments.

[5] Algorithm s-PA has been adapted, redefining the updating rule for the first and the last camera.

5

Fundamental Properties of Multi-Rotor UAVs

Aerial robotics is a fast-growing branch in robotics and Unmanned Aerial Vehicles (UAVs) are rapidly increasing in popularity. Thanks to their versatility in performing a wide variety of different tasks, aerial platforms are quickly becoming a mature technology exploited in military, industrial and civil context.

Independently from the application, in the real-word scenario, two fundamental properties are generally desirable for the used UAVs, namely the possibility to independently control the position and the attitude of the vehicles and the robustness to the loss of one or more motors. In this chapter, these fundamental capabilities are investigated formalizing them through the definition of some necessary algebraic conditions on the control force and control moment input matrices of generically-tilted multi-rotors.

The contents of this chapter are available in
Michieletto G., Ryll M., and Franchi A. Control of statically hoverable multi-rotor aerial vehicles and application to rotor-failure robustness for hexarotors. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2747–2752, 2017b;
Michieletto G., Ryll M., and Franchi A. Fundamental Actuation Properties of Multi-rotors: Force-Moment Decoupling and Fail-safe Robustness. https://hal.laas.fr/hal-01612602. 2017c.

5.1 Aerial Multi-Rotor Vehicles

A UAV is an aerial platform with no pilot on-board that can be remotely controlled by a human operator or can autonomously fly either following a pre-programmed flight plan or according to a more complex dynamic automation system (Gupta, Ghonge, and Jawandhiya (2013)).

In the last decade the human-controlled vehicles (usually called *drones*) have become very popular for a variety of purposes. However, the interest of the robotics community is more focused on the development of *autonomous* UAVs which may differ for number, typology and geometry of the actuators, i.e., of the propellers/rotors, but are all characterized by an on-board computer running the guidance control. In this context, a popular research direction concerns the control of swarms of such smart aerial platforms that are required to behave as cooperative multi-agent systems (this topic will be treated in the next chapter). To realize this task, high-performance capabilities and robustness represent key features that have to characterized each device composing the formation.

5.1.1 Literature Overview

Quadrotors (namely platforms with four actuators) constitute the most common autonomous UAV currently used in urban, rural, manufacturing, military and academic context. Their high versatility allows to their application field to range from exploration and mapping to grasping, from monitoring and surveillance to transportation (Duggal, Sukhwani, Bipin, Reddy, and Krishna (2016); Park and Kim (2016); Mellinger, Shomin, Michael, and Kumar (2013); Spica, Franchi, Oriolo, Bülthoff, and Giordano (2012)).

Besides their high popularity, standard quadrotors suffer of several limitations, such as, for instance, the limited payload capacity. The main drawback is however constituted by their under-actuated nature: they have to cope with six DoFs (three translational DoFs and three rotational DoFs), owning only four control inputs. This translates into the limitation of the executable manuvers set, and even into criticism to interact with the environment by exerting forces only in an arbitrarily-chosen direction of the space (Franchi, Carli, Bicego, and Ryll (2016)). To overcome this issue, the interest of robotics communities is now moving toward modeling, design and control of more complex multi-rotor platforms, where the number of propellers is larger than four (Rajappa, Ryll, Bülthoff, and Franchi (2015); Brescianini and D'Andrea (2016); Adîr, Stoica, and Whidborne (2013); Haddadi and Zarafshan (2015)).

Several hexarotor and octorotor vehicles (namely platforms with six and eight actuators, respectively) have been recently presented for applications spanning from multi-agent cooperative manipulation (see, e.g., Staub, Mohammadi, Bicego, Prattichizzo, and Franchi (2017) and the references therein) to human and environment interaction (see, e.g., Ryll, Muscio, Pierri, Cataldi, Antonelli, Caccavale, and Franchi (2017)). Intuitively, the intrinsic redundancy of these platforms can be exploited to enhance the actuation properties allowing to independently control the position. In addition, the presence of a greater number of actuators may improve also the robustness property of the platforms with respect to the loss of one or more propellers, which constitutes a key requirement for the real-world deployment.

In this direction, particular attention has been addressed to the six-rotor case and several recent works have presented new design solutions to ensure their full-actuation. These are mainly based on a tilt-rotor architecture, whose effectiveness has been exhaustively validated even considering quadrotor platforms (see, e.g., Ryll, Bülthoff, and Giordano (2012); Nemati and Kumar (2014)). Ryll, Bicego, and Franchi (2016) have shown that a standard star-shaped hexarotor can gain the 6-DoFs actuation using only one additional motor that allows to equally tilt all the propellers in a synchronized way.

Furthermore, it has been proven that in case of a rotor-loss the propellers mutual orientations affect the hexarotor control properties. For example, Du, Quan, Yang, and Cai (2015) have conducted a *controllability* analysis based on the observation that the dynamic model of a multi-rotor around hovering condition can be approximated by a linear system. Studying its algebraic properties, they have concluded that in case of a rotor failure the controllability strongly depends on the considered configuration in terms of the propeller spinning directions. Similarly, Mehmood, Nakamura, and Johnson (2016) have introduced and investigated the concept of *maneuverability* for a star-shaped hexarotor having tilted arms, when one propeller stops rotating. Maneuverability has been defined in terms of maximum acceleration achievable with respect to the six DoFs that characterize the dynamics of a UAV. In the failed-motor case, this reduces due to the loss in control authority and the hovering condition is still possible only for some tilt of the propellers. Giribet, Sanchez-Pena, and Ghersin (2016) have instead proposed a method to design a star-shaped hexarotor keeping the ability to reject disturbance torques in all directions while counteracting the effect of a failure in any motor. Their solution rests on (inward/outward) tilting all the propellers of a small fixed angle.

5.1.2 Problem Formulation

Motivated by the existing works and with the perspective of designing efficient UAVs formations, a generic class of UAVs in introduced in the following in order to figure out the relations between platforms design and performance/robustness properties.

A large part of the aerial platforms described in literature can be ideally modeled as *Generically Tilted Multi-Rotors* (GTMs). A GTM consists of a rigid body and n lightweight propellers. Each of them spins about its own axis (including the special cases of all parallel or all different axes) and is characterized by negligible mass, gyroscopic effect, and moment of inertia with respect to the body inertial parameters. The number n of propellers and the axes mutual orientations determine if the GTM is an under-actuated or full-actuated system (Rajappa et al. (2015)). Despite this, it has always full motion capabilities in the 3D space, as summarized in Figure 5.1.

To describe the dynamics of a GTM, the *body frame* $\mathscr{F}_B = \{O_B, (\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B)\}$ is introduced: the origin O_B coincides with the CoM of the platform, while $\mathbf{z}_B$-axis is down

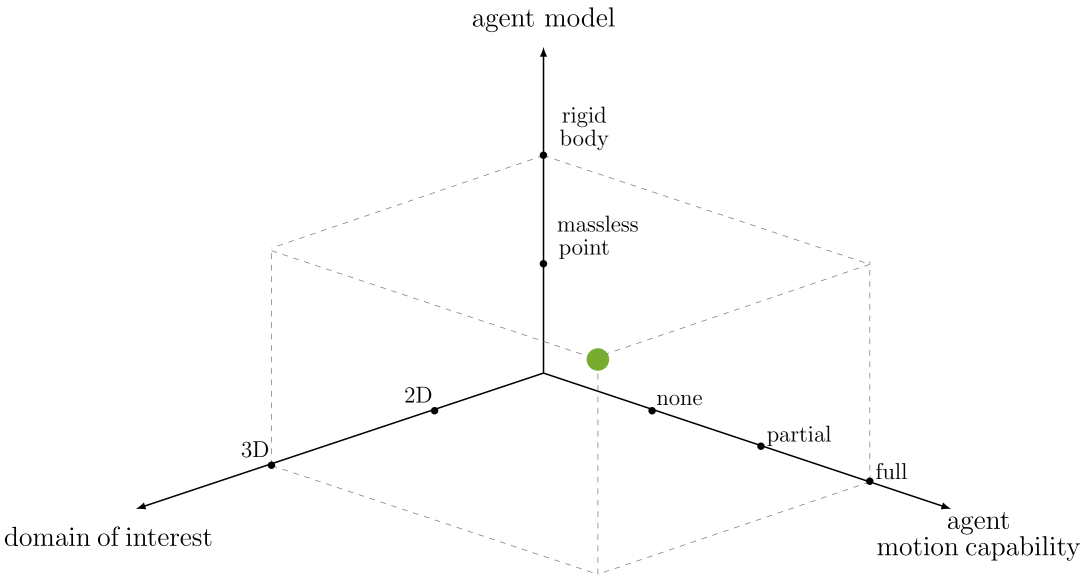


Figure 5.1. UAV: each GTM is modeled as a rigid body having full motion capability and acting in the 3D space.

faced oriented (contrarily to the camera case introduced in Chapter 3). The position of O_B in the inertial *world frame* $\mathscr{F}_W$ and the orientation of $\mathscr{F}_B$ with respect to $\mathscr{F}_W$ are respectively denoted by the vector $\mathbf{p} \in \mathbb{R}^3$ and by the rotation matrix $\mathbf{R} \in SO(3)$, hence the pair $\mathcal{X} = (\mathbf{p}, \mathbf{R}) \in \mathbb{R}^3 \times SO(3)$ describes the pose of the vehicle in $\mathscr{F}_W$. Note that the Cartesian product $\mathbb{R}^3 \times SO(3)$ identifies the so-called three-dimensional Special Euclidean group $SE(3)$ that describes the rigid-body motion in the 3D space.

The twist of the platform is indicated by the pair $(\mathbf{v}, \boldsymbol{\omega})$ where $\mathbf{v} = \mathbf{R}^\top \dot{\mathbf{p}} \in \mathbb{R}^3$ denotes the linear velocity of O_B in $\mathscr{F}_B$, and $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity of $\mathscr{F}_B$ with respect to $\mathscr{F}_W$, expressed in $\mathscr{F}_B$. Thus, the orientation kinematics is governed by the relation

$$\dot{\mathbf{R}} = \mathbf{R}[\boldsymbol{\omega}]_\times . \tag{5.1}$$

The motion equations of the GTM are derived simply using the standard Newton-Euler approach for the dynamics and considering the forces and torques that are generated by each propeller (Mahony and Kumar (2012)). Figure 5.2 highlights the involved quantities.

The i-th propeller, with $i = 1 \dots n$, rotates with an angular velocity $\boldsymbol{\omega}_i \in \mathbb{R}^3$ about a spinning axis which passes through the propeller center O_{P_i}. Both the direction of $\boldsymbol{\omega}_i$ and the position $\mathbf{p}_i \in \mathbb{R}^3$ of O_{P_i} are assumed constant in $\mathscr{F}_B$. According to the most commonly accepted model, the propeller applies at O_{P_i} a thrust (or lift) force $\mathbf{f}_i \in \mathbb{R}^3$ that is equal to

$$\mathbf{f}_i = \kappa c_{f_i} \|\boldsymbol{\omega}_i\| \boldsymbol{\omega}_i , \tag{5.2}$$

where $c_{f_i} \in \mathbb{R}^+$ is the norm of $\mathbf{f}_i$ when $\|\boldsymbol{\omega}_i\| = 1$ and $\kappa \in \{-1, 1\}$. Both c_{f_i} and κ are constant parameters depending on the shape of the rotor. The propeller is said of CCW type if $\kappa = 1$ and of CW type if $\kappa = -1$. For CCW propellers the lift has the same

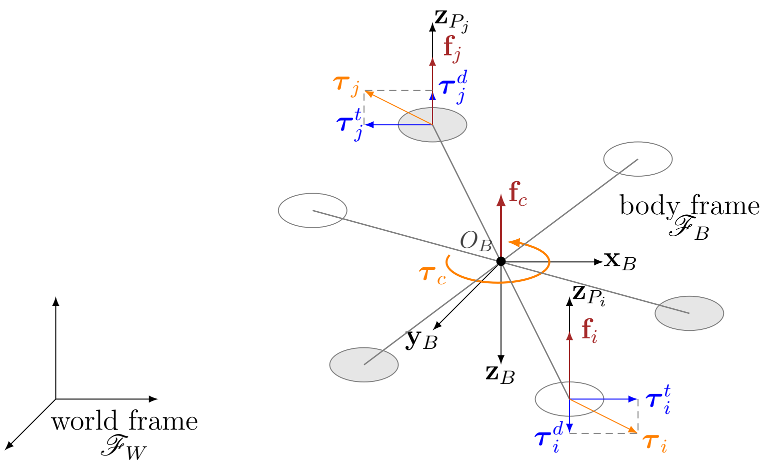


Figure 5.2. Instantiation of a GTM having six propeller: its dynamics depends on the implemented control force $\mathbf{f}_c$ and control moment $\boldsymbol{\tau}_c$.

direction of the angular velocity of the propeller, while for the CW it has the opposite direction. Moreover the i-th propeller generates a drag moment $\boldsymbol{\tau}_i^d \in \mathbb{R}^3$ whose direction is always opposite to its angular velocity, indeed

$$\boldsymbol{\tau}_i^d = -c_{\tau_i}^+ \|\boldsymbol{\omega}_i\| \boldsymbol{\omega}_i, \tag{5.3}$$

where $c_{\tau_i}^+ \in \mathbb{R}^+$ is the norm of $\boldsymbol{\tau}_i^d$ when $\|\boldsymbol{\omega}_i\| = 1$. Also $c_{\tau_i}^+$ is a constant parameter depending on the shape of the rotor.

One can arbitrarily choose a unit direction vector $\mathbf{z}_{P_i} \in \mathbb{R}^3$ that is parallel to the i-th propeller spinning axis, and that is also constant in $\mathscr{F}_B$. The angular velocity is then expressed as $\boldsymbol{\omega}_i = (\boldsymbol{\omega}_i^\top \mathbf{z}_{P_i})\mathbf{z}_{P_i} = \omega_i \mathbf{z}_{P_i}$. The scalar quantity $\omega_i \in \mathbb{R}$ is called the propeller spinning rate. Substituting this last expression in (5.2) and (5.3), it follows that

$$\mathbf{f}_i = \kappa c_{f_i} |\omega_i| \omega_i \mathbf{z}_{P_i} = c_{f_i} u_i \mathbf{z}_{P_i}, \tag{5.4}$$

$$\boldsymbol{\tau}_i^d = -c_{\tau_i}^+ |\omega_i| \omega_i \mathbf{z}_{P_i} = c_{\tau_i} u_i \mathbf{z}_{P_i}, \tag{5.5}$$

where $u_i = \kappa|\omega_i|\omega_i$ and $c_{\tau_i} = -\kappa c_{\tau_i}^+$. In this way the new input $u_i \in \mathbb{R}$, a one to one mapping with the propeller spinning rate, appears linearly in the force and moment equation. The type of propeller is understood by the sign of c_{τ_i} ($c_{\tau_i} < 0$ for the CCW type and $c_{\tau_i} > 0$ for CW type).

Denoting with $\boldsymbol{\tau}_i^t = \mathbf{p}_i \times \mathbf{f}_i \in \mathbb{R}^3$ the thrust moment associated to the i-th propeller[1], the total control force $\mathbf{f}_c \in \mathbb{R}^3$ and the total control moment $\boldsymbol{\tau}_c \in \mathbb{R}^3$ applied at O_B and expressed in $\mathscr{F}_B$ are

$$\mathbf{f}_c = \textstyle\sum_{i=1}^n \mathbf{f}_i = \sum_{i=1}^n c_{f_i} \mathbf{z}_{P_i} u_i, \tag{5.6}$$

$$\boldsymbol{\tau}_c = \textstyle\sum_{i=1}^n (\boldsymbol{\tau}_i^t + \boldsymbol{\tau}_i^d) = \sum_{i=1}^n \left(c_{f_i} \mathbf{p}_i \times \mathbf{z}_{P_i} + c_{\tau_i} \mathbf{z}_{P_i} \right) u_i. \tag{5.7}$$

[1] The propeller is assumed to be lightweight enough that the associated inertia moment can be neglected with respect to the other aerodynamic effects and platform inertia.

Introducing the control input vector $\mathbf{u} = [u_1 \ \cdots \ u_n]^\top \in \mathbb{R}^n$, (5.6) and (5.7) can be shortened as

$$\mathbf{f}_c = \mathbf{F}\mathbf{u}, \quad \text{and} \quad \boldsymbol{\tau}_c = \mathbf{M}\mathbf{u}, \tag{5.8}$$

where the *control force input matrix* $\mathbf{F} \in \mathbb{R}^{3\times n}$ and the *control moment input matrix* $\mathbf{M} \in \mathbb{R}^{3\times n}$ depend on the geometric and aerodynamic parameters introduced before.

The facts that $|c_{f_i}| > 0$ and $|c_{\tau_i}| > 0$ imply that none of the columns of both $\mathbf{F}$ and $\mathbf{M}$ is a zero vector, and therefore it holds both $\mathrm{rk}(\mathbf{F}) \geq 1$ and $\mathrm{rk}(\mathbf{M}) \geq 1$ by construction.

Neglecting the second order effects (such as, the gyroscopic and inertial effects due to the rotors and the flapping) the dynamics of the GTM is described by the following system of Newton-Euler equations

$$m\ddot{\mathbf{p}} = -mg\mathbf{e}_3 + \mathbf{R}\mathbf{f}_c = -mg\mathbf{e}_3 + \mathbf{R}\mathbf{F}\mathbf{u} \tag{5.9}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau}_c = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}\mathbf{u}, \tag{5.10}$$

where $g > 0$, $m > 0$ and $\mathbf{J} \in \mathbb{R}^{3\times 3}$ are the gravitational acceleration, the total mass of the platform and its positive definite inertia matrix, respectively, and $\mathbf{e}_i$ is the i-th canonical basis vector of $\mathbb{R}^3$ with $i = 1, 2, 3$.

According to (5.1) and (5.9)-(5.10), the dynamics of the angular velocity $\boldsymbol{\omega}$ influences the one of vehicle orientation $\mathbf{R}$. In turn, $\mathbf{R}$ influences the dynamics of $\dot{\mathbf{p}}$, and then of $\mathbf{p}$. The fact that this influence chain goes only in one way makes GTMs cascaded dynamical systems, a property that has been often used for controlling these kinds of platforms, see, e.g., Bouabdallah and Siegwart (2007). The cascaded dependency of the translational dynamics from the rotation one implies the emergence of a coupling between the control force and the control moment which may affect the performance of the platform. As a consequence, this interplay is investigated in the next section where some necessary algebraic conditions on the control input space are derived that imply the possibility to independently act on the vehicle position and attitude.

The rest of the chapter is then, devoted, to the analysis of another fundamental actuation property, namely the capability of a GTM platform to react after a propeller loss. The concept of rotor-failure robustness is formalized depending on the possibility for a multi-rotor to hover in a constant spot with zero linear and angular velocity (static hovering realizability property) even in case a propeller fails and stops spinning while being able to produce a full set of control inputs. Based on this definition, an extensive discussion on the robustness properties of the hexarotor platforms is carried out.

5.2 Force-Moment Decoupling Property

This section is devoted to the analysis of the coupling between the control force and the control moment that emerges from the intrinsic cascaded dependency of the UAV translational dynamics from the rotation one.

First, the considered GTM is assumed to be at least designed to satisfy

$$\mathrm{rk}(\mathbf{M}) = 3. \tag{5.11}$$

As a consequence of (5.11), the input space $\mathbb{R}^n$ can always be partitioned into the orthogonal subspaces $\mathrm{Im}\left(\mathbf{M}^\top\right)$ and $\mathrm{Im}\left(\mathbf{M}^\top\right)^\perp = \ker\left(\mathbf{M}\right)$, such that the vector $\mathbf{u}$ can be rewritten as the sum of two terms, namely

$$\mathbf{u} = \mathbf{T}\tilde{\mathbf{u}} = [\mathbf{A} \;\; \mathbf{B}] \begin{bmatrix} \tilde{\mathbf{u}}_A \\ \tilde{\mathbf{u}}_B \end{bmatrix} = \mathbf{A}\tilde{\mathbf{u}}_A + \mathbf{B}\tilde{\mathbf{u}}_B, \tag{5.12}$$

where $\mathbf{T} = [\mathbf{A} \;\; \mathbf{B}] \in \mathbb{R}^{n\times n}$ is an orthonormal matrix such that $\mathrm{Im}(\mathbf{A}) = \mathrm{Im}(\mathbf{M}^\top)$ and $\mathrm{Im}(\mathbf{B}) = \ker(\mathbf{M})$. Note that, due to (5.11), $\mathbf{A} \in \mathbb{R}^{n\times 3}$ is full rank, i.e., $\mathrm{rk}(\mathbf{A}) = 3$, while $\mathbf{B} \in \mathbb{R}^{n\times n-3}$ has $\mathrm{rk}(\mathbf{B}) = n-3$. Given this partition, it follows that

$$\boldsymbol{\tau}_c = \mathbf{M}\mathbf{T}_2\tilde{\mathbf{u}} = \mathbf{M}\mathbf{A}\tilde{\mathbf{u}}_A, \tag{5.13}$$

$$\mathbf{f}_c = \mathbf{F}\mathbf{T}\tilde{\mathbf{u}} = \mathbf{F}\mathbf{A}\tilde{\mathbf{u}}_A + \mathbf{F}\mathbf{B}\tilde{\mathbf{u}}_B = \mathbf{f}_c^A + \mathbf{f}_c^B. \tag{5.14}$$

Since, the matrix $\mathbf{MA}$ in (5.13) is non-singular, any moment $\boldsymbol{\tau} \in \mathbb{R}^3$ can be virtually implemented by setting $\tilde{\mathbf{u}}_A = (\mathbf{FA})^{-1}\boldsymbol{\tau}$ in conjunction with any $\tilde{\mathbf{u}}_B \in \mathbb{R}^{n-3}$.

The control force, which belongs to $\mathfrak{F} = \mathrm{Im}(\mathbf{F})$, is split in two components: $\mathbf{f}_c = \mathbf{f}_c^A + \mathbf{f}_c^B$. The component $\mathbf{f}_c^A = \mathbf{FA}\tilde{\mathbf{u}}_A$ represents the 'spurious' force generated by the allocation of the input, needed to obtain non-zero control moment. This component belongs to the subspace $\mathfrak{F}_A = \mathrm{Im}(\mathbf{FA})$. The component $\mathbf{f}_c^B = \mathbf{FB}\tilde{\mathbf{u}}_B$, instead, represents a force that can be assigned independently from the control moment by allocating the input $\mathbf{u}$ in $\mathrm{Im}(\mathbf{B}) = \ker(\mathbf{M})$. This 'free' force component belongs to the subspace $\mathfrak{F}_B = \mathrm{Im}(\mathbf{FB})$ and it is obtained by assigning $\tilde{\mathbf{u}}_B$. Being $\mathbf{T}$ non-singular, it follows that $\mathfrak{F} = \mathfrak{F}_A + \mathfrak{F}_B$. Recalling that $1 \leq \dim \mathfrak{F} \leq 3$ because $\mathrm{rk}(\mathbf{F}) \geq 1$, and that $\mathfrak{F}_B \subseteq \mathfrak{F}$, thus $\dim \mathfrak{F} \geq \dim \mathfrak{F}_B$.

The dimension of $\mathfrak{F}_B$ and its relation with $\mathfrak{F}$ sheds light upon the GTM actuation capabilities. The following two sets of definitions are devoted to these aspects.

Definition 5.2.1 (Decoupling properties)**.** A GTM is

- *fully coupled* (FC) if $\dim \mathfrak{F}_B = 0$ (i.e., if $\mathbf{FB} = \mathbf{0}$);
- *partially coupled* (PC) if $\dim \mathfrak{F}_B \in \{1, 2\}$ and $\mathfrak{F}_B \subset \mathfrak{F}$;
- *un-coupled* (UC) if $\mathfrak{F}_B = \mathfrak{F}$ (or, equivalently, $\mathfrak{F}_A \subseteq \mathfrak{F}_B$).

In a fully coupled GTM the control force depends completely upon the implemented control moment, in fact $\mathbf{f}_c^B = \mathbf{0}$ and thus $\mathbf{f}_c = \mathbf{f}_c^A$. In a partially coupled GTM the projection of the control force onto $\mathfrak{F}_B$ can be chosen freely while the projection onto $\mathfrak{F}_B^\perp \cap \mathfrak{F}$ depends completely upon the implemented control moment. Finally in an un-coupled (equivalently, fully decoupled) GTM no projection of the control force depends on the control moment, i.e., the control force can be freely assigned in the whole space $\mathfrak{F}$.

Note that the full decoupling does not imply necessarily that the control force can be chosen in the whole $\mathbb{R}^3$, unless it holds also $\mathfrak{F} = \mathbb{R}^3$.

Definition 5.2.2 (Actuation properties)**.** A GTM

- has a *preferential direction* (1P) if $\dim \mathfrak{F}_B \geq 1$;
- has a *preferential plane* (2P) if $\dim \mathfrak{F}_B \geq 2$;
- is *fully actuated* (3P) if $\dim \mathfrak{F}_B = 3$.

Combining the previous definitions, it holds that a GTM

- has a *single preferential direction* (S-1P) if $\dim \mathfrak{F}_B = 1$;
- has a *single preferential plane* (S-2P) if $\dim \mathfrak{F}_B = 2$.

If a GTM has a preferential direction then there exists at least a direction along which the projection of the control force can be chosen freely from the control moment. If a GTM has a preferential plane then there exists at least a plane over which the projection of the control force can be chosen freely from the control moment. If a GTM is fully actuated then the control force can be chosen in all $\mathbb{R}^3$ freely from the control moment.

Remark 5.2.3. It is possible to show that the above definition of full-actuation is equivalent to the more common definition known in the literature (see, e.g., Brescianini and D'Andrea (2016)), i.e.,

$$\mathrm{rk}(\mathbf{W}) = \mathrm{rk}\left(\begin{bmatrix}\mathbf{F}\\ \mathbf{M}\end{bmatrix}\right) = 6. \tag{5.15}$$

Post-multiplying $\mathbf{W}$ by $\mathbf{T}$ does not change the rank and it occurs that

$$\mathbf{WT} = \begin{bmatrix}\mathbf{F}\\ \mathbf{M}\end{bmatrix}[\mathbf{A}\ \mathbf{B}] = \begin{bmatrix}\mathbf{FA} & \mathbf{FB}\\ \mathbf{MA} & \mathbf{0}\end{bmatrix}. \tag{5.16}$$

Because $\mathrm{rk}(\mathbf{MA}) = 3$ thanks to (5.11), it follows that $\mathrm{rk}(\mathbf{W}) = 6$ if and only if $\mathrm{rk}(\mathbf{FB}) = 3$, which corresponds to the FA definition given above.

In terms of relations between the above definitions, one can note that: 3P implies UC, while the converse is not true; 3P implies 2P; 2P implies 1P. Finally, 1P (and thus 2P) can coexist with PC or UC but not with FC.

Table 5.1 yields a comprehensive view of all the aforementioned definitions and relations. In the following two illustrative examples of GMT are provided: their coupling properties are evaluated using the tools just provided.

		∃ preferential direction		
			∃ preferential plane	
	$\dim \mathfrak{F}_B = 0$	$\dim \mathfrak{F}_B = 1$	$\dim \mathfrak{F}_B = 2$	$\dim \mathfrak{F}_B = 3$
$\mathfrak{F}_B \subsetneq \mathfrak{F}$	FC	PC and S-1P	PC and S-2P	N/A
$\mathfrak{F}_B = \mathfrak{F}$	N/A	UC and S-1P	UC and S-2P	3P (UC)
	$(\dim \mathfrak{F} \geq 1)$	$(\dim \mathfrak{F} \geq 1)$	$(\Rightarrow \dim \mathfrak{F} \geq 2)$	$(\Rightarrow \dim \mathfrak{F} = 3)$

Table 5.1. A table recalling the fundamental properties of the actuation of a GTM.

5.2.1 Standard (collinear) Multi-Rotors

Consider the case in which $\mathrm{Im}(\mathbf{F}^\top) \subseteq \ker(\mathbf{M}) = \mathrm{Im}(\mathbf{B})$. By definition $\mathbf{F} \neq \mathbf{0}$, hence this hypothesis implies that $\mathbf{FB} \neq \mathbf{0}$ and that

$$\mathbf{MF}^\top = \mathbf{0} \Leftrightarrow \mathbf{FM}^\top = \mathbf{0} \Leftrightarrow \mathbf{FA} = \mathbf{0}. \tag{5.17}$$

Therefore $\mathfrak{F}_A = \{\mathbf{0}\}$ and thus $\mathfrak{F}_B = \mathfrak{F}$, i.e., the GMT is UC. Another way to see it is to note that $\mathbf{f}_c = \mathbf{FB}\tilde{\mathbf{u}}_B$ according to (5.14) and hence $\mathbf{f}_c$ is independent of $\tilde{\mathbf{u}}_A$, i.e., of $\boldsymbol{\tau}_c$.

Classical multi-rotor systems fall in this case. They are characterized by an even number of propellers having parallel orientations ($\mathbf{z}_{P_i} = \mathbf{z}_P \; \forall i$), a balanced geometry and a balanced choice of CW/CCW spinning directions. Their matrices $\mathbf{F}$ and $\mathbf{M}$ result to be

$$\begin{aligned} \mathbf{F} &= \begin{bmatrix} c_{f_1}\mathbf{z}_P & \cdots & c_{f_n}\mathbf{z}_P \end{bmatrix}, \\ \mathbf{M} &= \begin{bmatrix} c_{f_1}\mathbf{p}_1 \times \mathbf{z}_P & \cdots & c_{f_n}\mathbf{p}_n \times \mathbf{z}_P \end{bmatrix} + \begin{bmatrix} c_{\tau_1}\mathbf{z}_P & \cdots & c_{\tau_n}\mathbf{z}_P \end{bmatrix}. \end{aligned} \tag{5.18}$$

Notice, to have $\mathrm{rk}(\mathbf{M}) = 3$ it is enough to choose at least the position vectors of two propellers i and j such that $\mathbf{p}_i \times \mathbf{z}_P$, $\mathbf{p}_j \times \mathbf{z}_P$, and $\mathbf{z}_P$ are linearly independent.

To show that $\mathbf{MF}^\top = \mathbf{0}$, it has first to be observed that $\mathbf{MF}^\top = \mathbf{C}_f + \mathbf{C}_\tau$, where

$$\mathbf{C}_f = \left(\left(\textstyle\sum_{i=1}^n c_{f_i}^2 \mathbf{p}_i\right) \times \mathbf{z}_P\right) \mathbf{z}_P^\top \in \mathbb{R}^{3\times 3}, \tag{5.19}$$

$$\mathbf{C}_\tau = \left(\textstyle\sum_{i=1}^n c_{\tau_i} c_{f_i}\right) \mathbf{z}_P \mathbf{z}_P^\top \in \mathbb{R}^{3\times 3}. \tag{5.20}$$

Then, by suitably choosing the positions and the coefficients $\{c_{\tau_i}, c_{f_i}\}$ one can easily make $\mathbf{C}_f = \mathbf{C}_\tau = \mathbf{0}$. For example it is enough to make the propellers pairwise balanced, i.e., satisfying $\mathbf{p}_i + \mathbf{p}_j = \mathbf{0}$, $c_{f_i} = c_{f_j}$, and $c_{\tau_i} = -c_{\tau_j}$ for $i \in \{1 \ldots \frac{n}{2}\}$ and $j = i + \frac{n}{2}$. Many other choices are however possible.

Finally, with respect to Table 5.1, note that such a multi-rotor system has also a preferential direction but not a preferential plane, because $\mathrm{rk}(\mathbf{F}) = 1$ and thus $\dim \mathfrak{F}_B = 1$. Classical multi-rotor systems are therefore fully decoupled GTMs with a single preferential direction. In these platforms control moment and control force can be independently considered however the control force is always oriented in the same direction regardless of the value of the input $\mathbf{u}$ so that this is not affected by the unavoidable uncertainty of the input. On the contrary, the direction can be reliably measured by simple attitude estimation, as well as its derivative (by a gyroscope) and controlled through the fully actuated rotational dynamics. All these properties are fundamental to establish the success and simplicity in controlling such platforms. The only price to pay is underactuation, which has not been an obstacle in many cases of practical relevance.

5.2.2 Tilted Quadrotor

The tilted quadrotor used in the experimental setup in Falanga, Mueggler, Faessler, and Scaramuzza (2017) constitutes an example of a platform which is instead partially coupled with a single preferential direction. This vehicle is such that the i-th propeller is tilted about the axis joining O_B with O_{P_i} of an angle α_i in a way that the consecutive rotors are oriented in opposite way, i.e., $\alpha_1 = \alpha_3 = \alpha$ and $\alpha_2 = \alpha_4 = -\alpha$, with $\alpha \in \left[0, \frac{\pi}{2}\right]$. Hence, assuming that all the propellers have the same aerodynamic features (namely $c_{f_i} = c_f$ and $|c_{\tau_i}| = c_\tau$, with $i = 1 \ldots 4$), it is possible to derive

$$\mathbf{F} = c_f \begin{bmatrix} 0 & s\alpha & 0 & -s\alpha \\ s\alpha & 0 & -s\alpha & 0 \\ c\alpha & c\alpha & c\alpha & c\alpha \end{bmatrix} \tag{5.21}$$

$$\mathbf{M} = c_\tau \begin{bmatrix} 0 & s\alpha + rc\alpha & 0 & -s\alpha - rc\alpha \\ -s\alpha - rc\alpha & 0 & s\alpha + rc\alpha & 0 \\ -c\alpha + rs\alpha & c\alpha - rs\alpha & -c\alpha + rs\alpha & c\alpha - rs\alpha \end{bmatrix} \tag{5.22}$$

where $r = (c_f/c_\tau)l$ with $l \in \mathbb{R}^+$ denoting the distance between O_B and O_{P_i}, and $s\alpha = \sin\alpha$ and $c\alpha = \cos\alpha$.

From (5.21) it is easy to see that $\mathfrak{F} = \mathbb{R}^3$ if $s\alpha \neq 0$ and $c\alpha \neq 0$, while $\mathfrak{F} = \operatorname{span}\{\mathbf{e}_3\}$ if $s\alpha = 0$, and finally $\mathfrak{F} = \operatorname{span}\{\mathbf{e}_1, \mathbf{e}_2\}$ if $c\alpha = 0$. In addition, $\mathbf{M}$ in (5.22) results to be full rank if $\tan\alpha \neq -r$ and $\tan\alpha \neq \frac{1}{r}$, whereas if $\tan\alpha = \frac{1}{r}$ $(-c\alpha + rs\alpha = 0)$ then $\operatorname{rk}(\mathbf{M}) = 2$ and if $\tan\alpha = -r$ $(s\alpha + rc\alpha = 0)$ then $\operatorname{rk}(\mathbf{M}) = 1$. When $\operatorname{rk}(\mathbf{M}) = 3$, according to (5.12), the input space $\mathbb{R}^4$ can be partitioned by choosing, for example,

$$\mathbf{A} = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}. \tag{5.23}$$

As a consequence, it follows

$$\mathbf{FA} = 2c_f \begin{bmatrix} s\alpha & 0 & 0 \\ 0 & -s\alpha & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{FB} = 4c_f \begin{bmatrix} 0 \\ 0 \\ c\alpha \end{bmatrix}. \tag{5.24}$$

When $c\alpha = 0$ the GTM is FC because $\dim \mathfrak{F}_B = 0$. Instead, as long as $c\alpha \neq 0$, it is $\dim \mathfrak{F}_B = \operatorname{Im}(\mathbf{FB}) = 1$, i.e., the GTM has a single preferential direction, which is $\mathbf{e}_3$. In this case the platform is UC if and only if $s\alpha = 0$, in fact only in this case $\mathfrak{F}_B = \mathfrak{F}$ (or equivalently $\mathfrak{F}_A = \{\mathbf{0}\} \subseteq \mathfrak{F}_B$). In the case in which $s\alpha \neq 0$ (as in Falanga et al. (2017)) the GMT is PC. The plane $\mathfrak{F}_B^{\perp} \cap \mathfrak{F} = \text{span}\{\mathbf{e}_1, \mathbf{e}_2\}$ represents the plane along which the projection of the control force depends completely on the choice of the control moment. In Falanga et al. (2017), the effect of this term is partially mitigated by the robustness of the hovering controller, however the perfect tracking that is possible with $\alpha = 0$ is theoretically not guaranteed anymore.

5.3 Static Hovering with Unidirectional Propeller Spin

The large majority of propellers used in GTMs can spin only in one direction, mainly due to the larger efficiency of rotors with asymmetric profile and the difficulty in reliably and quickly changing the spinning direction. It is therefore important to consider this additional constraint in the UAV model and evaluate the consequences.

In the rest of the section, the goal is to analyze the conditions under which a GTM can stay in a controlled static equilibrium when the additional constraint $\mathbf{u} \geq \mathbf{0}$ is enforced. To this aim, the following definition is provided.

Definition 5.3.1 (Equilibrium)**.** A GTM is in equilibrium if

$$\dot{\mathbf{p}} = \mathbf{0}, \quad \ddot{\mathbf{p}} = \mathbf{0}, \quad \boldsymbol{\omega} = \mathbf{0}, \quad \dot{\boldsymbol{\omega}} = \mathbf{0} \qquad \text{or, equivalently} \tag{5.25}$$

$$\dot{\mathbf{p}} = \mathbf{0}, \quad \mathbf{f}_c = \mathbf{F}\mathbf{u} = mg\mathbf{R}^\top \mathbf{e}_3, \quad \boldsymbol{\omega} = \mathbf{0}, \quad \boldsymbol{\tau}_c = \mathbf{M}\mathbf{u} = \mathbf{0}. \tag{5.26}$$

A basic property to ensure the rejection of external disturbances while being in equilibrium is the possibility to exert a control moment $\boldsymbol{\tau}_c$ in any direction and with any intensity by a suitable allocation of the input vector $\mathbf{u} \geq \mathbf{0}$. In this perspective, in Giribet et al. (2016) the next condition has been introduced.

Definition 5.3.2 (Realizability of any control moment (Giribet et al. (2016)))**.** A GTM can realize any control moment if it is possible to allocate the actuator values $\mathbf{u} \geq \mathbf{0}$ to obtain any $\boldsymbol{\tau}_c \in \mathbb{R}^3$. Formally if

$$\forall \boldsymbol{\tau}_c \in \mathbb{R}^3 \quad \exists \mathbf{u} \geq \mathbf{0} \quad \text{s.t.} \quad \mathbf{M}\mathbf{u} = \boldsymbol{\tau}_c. \tag{5.27}$$

In Giribet et al. (2016) it has been shown that (5.27) is equivalent to the simultaneous satisfaction of (5.11) and the following condition

$$\exists \mathbf{u} > \mathbf{0} \quad \text{s.t.} \quad \mathbf{M}\mathbf{u} = \mathbf{0}. \tag{5.28}$$

A drawback of Definition 5.3.2 is that it considers only the realizability with respect to the generation of the control moment, thus ignoring the control force. However, a proper control force generation is also needed to robustly control the GTM while in equilibrium. For this reason, the following additional condition is given.

Definition 5.3.3 (Realizability of any control force)**.** A GTM can realize any control force if it is possible to allocate the actuator values $\mathbf{u} \geq \mathbf{0}$ to obtain a control force with any intensity $f_c \in \mathbb{R}_0^+$ while the platform is in static hovering. Formally if

$$\forall f_c \in \mathbb{R}_0^+ \quad \exists \mathbf{u} \geq \mathbf{0} \;\text{ s.t. }\; \mathbf{M}\mathbf{u} = \mathbf{0} \;\text{ and }\; \|\mathbf{F}\mathbf{u}\|_2 = f_c. \tag{5.29}$$

Note that the static hovering equilibrium (5.26) does not force the vehicle in a certain orientation. As a consequence, when it is possible to generate a control force with any non-negative intensity, then it is sufficient to attain the suitable attitude in order to realize any other control force vector.

Proposition 5.3.4. *Condition* (5.29) *is equivalent to*

$$\exists \mathbf{u} \geq \mathbf{0} \quad \textit{s.t.} \quad \mathbf{M}\mathbf{u} = \mathbf{0} \;\textit{ and }\; \mathbf{F}\mathbf{u} \neq \mathbf{0}. \tag{5.30}$$

Proof. The proof is straightforward and reported here only for completeness.

(5.30) ⇒ (5.29): Assume that $\bar{\mathbf{u}}$ satisfies (5.30), i.e., $\mathbf{M}\bar{\mathbf{u}} = \mathbf{0}$ and $\mathbf{F}\bar{\mathbf{u}} \neq \mathbf{0}$, then, for any $f_c \in \mathbb{R}_0^+$ there exists the vector $\mathbf{u} = f_c \bar{\mathbf{u}}/\|\mathbf{F}\bar{\mathbf{u}}\|_2$ which satisfies (5.29).

(5.29) ⇒ (5.30): Consider any $f_c \in \mathbb{R}^+$, and assume that $\mathbf{u}$ satisfies (5.29), then the same $\mathbf{u}$ satisfies also (5.30). □

Exploiting the previous equivalent conditions the following more complete definition is introduced for the realizability of the static hovering.

Definition 5.3.5 (Static hovering realizability)**.** If the three conditions (5.11), (5.28), and (5.30) are met, then the GTM can hover statically (with non-negative inputs).

Notice that (5.11), (5.28), and (5.30) are only necessary conditions for the equilibrium in Definition 5.3.1. The property of realizability of static hovering is indeed agnostic with respect to the set of attitudes at which this static hovering can be realized. These orientations are all those represented by a matrix $\mathbf{R}$ for which (5.26) holds with $\mathbf{u} \geq \mathbf{0}$. If a GTM can hover statically at least an attitude of such kind exists.

All the common star-shaped multi-rotors are GTM that can hover statically, as stated in the following proposition.

Proposition 5.3.6. *Multi-rotors having n propellers with $n \geq 4$ and even, such that*

- $c_{\tau_i} = c_\tau \in \mathbb{R}^+$ *for* $i = 1, 3 \ldots n-1$, $c_{\tau_i} = -c_\tau$ *for* $i = 2, 4 \ldots n$, *and* $c_{f_i} = c_f \in \mathbb{R}^+$,
- $\mathbf{z}_{P_i} = \mathbf{e}_3$ *and* $\mathbf{p}_i = l\,\mathbf{R}^z\left((i-1)\frac{2\pi}{n}\right)\mathbf{e}_1$ *for* $i = 1 \ldots n$ *where* $l \in \mathbb{R}^+$,

can realize static hovering.

Proof. After some simple algebra it is easy to check that $\mathbf{M}$ is full rank. Furthermore it is also easy to verify that the vector of all ones $\mathbf{1} = [1 \;\cdots\; 1]^\top \in \mathbb{R}^n$ has the property that $\mathbf{M}\mathbf{1} = \mathbf{0}$ and $\mathbf{F}\mathbf{1} \neq \mathbf{0}$, thus $\mathbf{u} = \mathbf{1}$ satisfies all the required conditions. □

Standard star-shaped multi-rotors described in Proposition 5.3.6 are not the only statically hoverable GTMs. In fact, in the next section other examples are shown that arise in the important situations of propeller failures. Conversely, it is also easy to find examples of GTMs that cannot hover statically, like the following one.

Proposition 5.3.7. *Consider a 4-rotor that respects all the conditions in Proposition 5.3.6 apart from the fact that $c_{\tau_i} = c_\tau > 0$ for $i = 1, 2$ and $c_{\tau_i} = -c_\tau$ for $i = 3, 4$. This GTM cannot realize static hovering.*

Proof. Expanding (5.7) for this special case, and noting that $\mathbf{p}_3 = -\mathbf{p}_1$ and $\mathbf{p}_4 = -\mathbf{p}_2$, it follows that

$$\begin{aligned}\boldsymbol{\tau}_c &= \textstyle\sum_{i=1}^n \left(c_{f_i}\mathbf{p}_i \times \mathbf{z}_{P_i} + c_{\tau_i}\mathbf{z}_{P_i}\right) u_i && (5.31)\\ &= \left(c_f\mathbf{p}_1{\times}\mathbf{e}_3 + c_\tau\mathbf{e}_3\right)(u_1 - u_3) + \left(c_f\mathbf{p}_2{\times}\mathbf{e}_3 + c_\tau\mathbf{e}_3\right)(u_2 - u_4). && (5.32)\end{aligned}$$

Denoting with $\mathbf{f}_{21} = (c_f\mathbf{p}_1{\times}\mathbf{e}_3 + c_\tau\mathbf{e}_3)$ and $\mathbf{f}_{22} = (c_f\mathbf{p}_2 \times \mathbf{e}_3 + c_\tau\mathbf{e}_3)$, then it occurs $\mathbf{M} = [\mathbf{f}_{12}\ \ \mathbf{f}_{22}\ \ -\mathbf{f}_{12}\ \ -\mathbf{f}_{22}]$ whose rank is 2 and therefore condition (5.11) is not met. □

Remark 5.3.8. If there was no constraint $\mathbf{u} \geq \mathbf{0}$ the capability of static hovering would have been equivalent to the existence of a preferential direction, while since the additional constraint $\mathbf{u} \geq \mathbf{0}$ is imposed one needs stronger properties to be fulfilled.

This remark is in line with the fact that GMTs which can hover statically have a preferential direction (see Definiton 5.2.2) as stated in the next proposition, but are not necessarily fully decoupled.

Proposition 5.3.9. *A GTM that can realize static hovering has a preferential direction. In particular, consider any $\mathbf{u} = \bar{\mathbf{u}} \in \mathbb{R}^n$ which satisfies (5.30), then a possible preferential direction is*

$$\mathbf{d}_* = \frac{\mathbf{F}\bar{\mathbf{u}}}{\|\mathbf{F}\bar{\mathbf{u}}\|_2}. \tag{5.33}$$

Proof. Since $\bar{\mathbf{u}} \in \ker(\mathbf{M})$, the rightmost requirement in (5.30) can be written as $\mathbf{F}\mathbf{B}\tilde{\mathbf{u}}_B \neq \mathbf{0}$, which implies $\dim \mathfrak{F}_B \geq 1$. □

5.3.1 Hovering Commands and Input Feasibility Condition

The conditions in Definition 5.3.5 take into account only the geometry of the given GTM, neglecting its physical features and limitations, that are conversely considered in the notion of *hovering feasibility* here introduced.

First, note that the conditions $\mathbf{M}\mathbf{u} = \mathbf{0}$ and $\mathbf{R}\mathbf{F}\mathbf{u} = mg\mathbf{e}_3$ required at the static hovering equilibrium (5.26) are both satisfied if one sets $\mathbf{u} = \delta\mathbf{u}_0$ with $\delta = mg/\|\mathbf{F}\mathbf{u}_0\|_2 \in \mathbb{R}^+$, $\mathbf{u}_0 \in \ker(\mathbf{M})$, and $\mathbf{R}$ is such that the vector $\mathbf{R}\mathbf{F}\mathbf{u}_0$ is directed along the direction identified by $\mathbf{e}_3$. In this way any vector $\mathbf{u}_0$ satisfying the conditions for the static hovering realizability geometrically represents a line in the input space $\mathbb{R}^n$, along which a command $\mathbf{u}$ can lie to keep the platform in hovering. The working point along this line is decided by the mass m of the platform and the propellers characteristics and arrangement, through the matrices $\mathbf{F}$ and $\mathbf{M}$, i.e., by the physical features of the platform.

In real normal working condition the i-th propeller spinning rate ω_i must belong to a certain set of feasible frequencies, namely the (non-negative) commands input are bounded so that $u_i \in \mathcal{U}_i$, $i = 1 \ldots n$. Taking care of these physical limitations of the motors, the static hovering condition is realizable whenever the working point input commands corresponding to a hovering equilibrium lie 'inside enough' the allowable input set $\mathcal{U}_1 \times \ldots \times \mathcal{U}_n$, such that additional input maneuverability is left for controlling the attitude and compensating the position errors. This physical feasibility requirement can be formally stated as in the following definition, wherein the conservative hovering commands set $\mathcal{H}_1 \times \ldots \times \mathcal{H}_n$ with $\mathcal{H}_1 \subset \mathcal{U}_1 \ldots \mathcal{H}_n \subset \mathcal{U}_n$ are introduced.

Definition 5.3.10 (Static hovering feasibility)**.** Static hovering is a physically feasible flight condition for a GTM if it is compliant to Definition 5.3.5 and the hovering commands $\delta\mathbf{u}_0$ belongs to $\mathcal{H}_1 \times \ldots \times \mathcal{H}_n$.

The conditions in Definition 5.3.10 can be slightly relaxed assuming that a subset of rotors with indexes in a set $\mathcal{C} \subset \mathcal{R} = \{1 \ldots n\}$ is not used for dynamic control but is just kept at a *constant spinning rate* for the whole duration of the flight, i.e., $u_j(t) = u_j^* \quad \forall j \in \mathcal{C}, \forall t \geq t_0$. In this case the physical feasibility requirement becomes less stringent, in fact it can be tolerated that $\delta u_{0,j} = u_j^*$ belongs to the larger set $\mathcal{U}_j$ (instead of $\mathcal{H}_j$) for any $j \in \mathcal{C}$, where $u_{0,j}$ represents the j-th entry of $\mathbf{u}_0$. On the other side by doing so only the control inputs u_i with $i \in \mathcal{R} \backslash \mathcal{C}$ can be used for dynamic control. Denoting with $\mathbf{F}^{\mathcal{C}}$, $\mathbf{M}^{\mathcal{C}}$, and $\mathbf{u}_0^{\mathcal{C}}$ the matrices (vector) obtained from $\mathbf{F}$, $\mathbf{M}$, ($\mathbf{u}_0$) by removing the columns (entries) whose index is in $\mathcal{C}$, the following conditions can be stated.

Definition 5.3.11 (Relaxed static hovering feasibility)**.** Static hovering is a physically feasible flight condition for a GTM if there exist $\mathcal{C} \subset \mathcal{R}$ and a set of values $\{u_j^* \in \mathcal{U}_j\}_{j \in \mathcal{C}}$ such that the following conditions are satisfied

1. $rk(\mathbf{M}^{\mathcal{C}}) = 3$,
2. $\exists \mathbf{u} > \mathbf{0} \quad \text{s.t.} \quad \mathbf{M}\mathbf{u} = \mathbf{0}$,
3. $\exists \mathbf{u}_0 \geq \mathbf{0} \quad \text{s.t.}$
   a) $\mathbf{M}\mathbf{u}_0 = \mathbf{0}$
   b) $\mathbf{F}^{\mathcal{C}}\mathbf{u}_0^{\mathcal{C}} \neq \mathbf{0}$
   c) denoting with $\delta = mg/\|\mathbf{F}\mathbf{u}_0\|$ it is $\delta u_{0,j} = u_j^*$ $\forall j \in \mathcal{C}$ and $\delta u_{0,j} \in \mathcal{H}_j$ for any other j.

Proposition 5.3.12. *A straightforward consequence of the Definition 5.3.11 is that static hovering is a physically feasible flight condition for a GTM only if $n - |\mathcal{C}| \geq 4$.*

Furthermore, if all the propellers are the same, i.e., $c_{f_i} = c_f \in \mathbb{R}^+$ and $\mathcal{U}_i = [\underline{u}, \overline{u}] \subset \mathbb{R}_0^+$ $\forall i = 1 \ldots n$, then it is clear that the more the propellers evenly share the burden to counterbalance the gravity force the more likely they will remain within their bounds. Therefore it becomes desirable to have $\mathbf{u}_0$ as much as possible aligned with $\mathbb{1}_n$, namely the bisector of the positive orthant in $\mathbb{R}^n$.

The notion of static hovering feasibility will be recalled in the next chapter, whereas in the following Definition 5.4.3 is considered in order to provide more general results.

5.4 Hexarotor Robustness

In this section the theory developed so far is applied to investigate the *rotor-failure robustness* of hexarotor GTMs (i.e., GTMs with $n = 6$). Robustness is defined as the capability of the platform to realize static hovering even in case a propeller fails and stops to spin. The attention is focused on platforms having six rotors because in Scaramuzza, Achtelik, Doitsidis, Friedrich, Kosmatopoulos, Martinelli, Achtelik, Chli, Chatzichristofis, Kneip, et al. (2014), it has been shown that it is the minimum number of actuators which guarantees the resolution of the controller allocation problem with redundancy against a single failure.

Definition 5.4.1 (Healthy and failed propellers)**.** In the following, 'the i-th rotor is *failed*' means that it stops to spin ($\omega_i = u_i = 0$), thus producing neither thrust nor drag anymore. A rotor that is not failed is *healthy*.

Definition 5.4.2 ($\{k\}$-loss robustness)**.** Given a hexarotor GTM whose propellers set is denoted by $\mathcal{P} = \{1 \dots 6\}$, this is said to be *$\{k\}$-loss robust* with $k \in \mathcal{P}$ if the pentarotor GTM obtained considering only the healthy rotors in $\mathcal{P}\backslash\{k\}$ can still realize static hover (according to Definition 5.3.5).

Definition 5.4.3 (Robustness properties)**.** A hexarotor GTM is said to be

- *fully robust* if it is $\{k\}$-loss robust for *any* $k \in \mathcal{P}$;
- *partially robust* if it is not fully robust but it is $\{k\}$-loss robust for *at least one* $k \in \mathcal{P}$;
- *fully vulnerable* if it is neither fully nor partially robust.

5.4.1 (α, β, γ)–Hexarotor Family

In the following, a fairly general hexarotor GTM model parametrized by three angles, α, β, and γ, is described. The angle γ is meant to define the propellers arrangement, while the angles α and β allow to describe the orientation of the rotors spinning direction as formally explained in the following. The corresponding (α, β, γ)*-hexarotor family* spans (and extends) the most commonly used classes of 6-rotor GTMs. The goal is to analyze the relation that exists between these angles and the robustness features of the members of this family. By doing so, the results presented in Giribet et al. (2016) are extended, since in such work only a family parametrized by β is considered (i.e., it is assumed $\alpha = \gamma = 0$) and only the compliance with Definition 5.3.2 is evaluated.

For a (α, β, γ)-hexarotor GTM the positions in $\mathscr{F}_B$ of the propeller centres O_{P_i}'s are given by

$$\mathbf{p}_i = l\,\mathbf{R}^z\left((i-1)\tfrac{\pi}{3} - \tfrac{1}{2}(1+(-1)^i)\gamma\right)\mathbf{e}_1 = l\,\mathbf{R}_\gamma(i)\mathbf{e}_1, \quad \forall i \in \mathcal{P}, \tag{5.34}$$

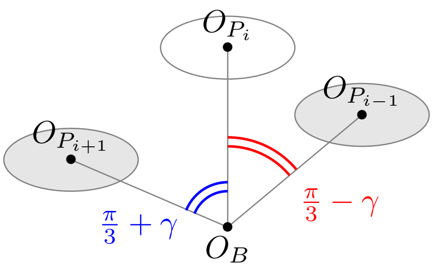


Figure 5.3. Position of three consecutive propellers in a (α,β,γ)-hexarotor GTM, highlighting the effect of the γ angle.

where $\gamma \in \left[0, \frac{\pi}{3}\right]$ and $l = \text{dist}(O_B, O_{P_i}) \in \mathbb{R}^+$. In this way the smallest angle between $\overline{O_B O_{P_i}}$ and $\overline{O_B O_{P_j}}$, with $j = (i \bmod 6) + 1$ is alternatively $\frac{\pi}{3} - \gamma$ and $\frac{\pi}{3} + \gamma$, as shown in Figure 5.3. Note that because of (5.34) all the propeller centres belong to the same plane that usually coincides with the plane identified by $\mathbf{x}_B$-axis and $\mathbf{y}_B$-axis.

The orientation of the i-th propeller is instead provided by

$$\mathbf{z}_{P_i} = \mathbf{R}_\gamma(i) \underbrace{\mathbf{R}^y(\beta)}_{\mathbf{R}_\beta} \underbrace{\mathbf{R}^x(\alpha_i)}_{\mathbf{R}_\alpha(i)} \mathbf{e}_3 = \mathbf{R}_{\alpha\beta\gamma}(i)\mathbf{e}_3, \tag{5.35}$$

where $\alpha_i = (-1)^{i-1}\alpha$ (with $i \in \mathcal{P}$), and $\alpha, \beta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$. To geometrically understand the meaning of (5.35) one can note that the unit vector $\mathbf{z}_{P_i}$ is equal to the z-axis of the frame obtained after the following two consecutive rotations applied to $\mathscr{F}_B$: the first is a rotation of an angle α_i about the vector $\overrightarrow{O_B O_{P_i}}$, while the second is a rotation of an angle β about the y-axis of the intermediate frame obtained after the first rotation.

In terms of aerodynamic coefficients, each hexarotor of the family has the following pattern

$$c_{f_i} = c_f, \quad c_{\tau_i} = (-1)^{i-1} c_\tau, \quad \forall i \in \mathcal{P}, \tag{5.36}$$

where c_f and c_τ are two constant values depending on the used propellers.

In the following, the most relevant configurations that can be obtained by changing the three angles are commented on. First, when sweeping γ from 0 to $\frac{\pi}{3}$ it takes place a smooth transition between the two most popular propeller arrangements for hexarotors depicted in Figure 5.4, i.e.,

- $\gamma = 0$: the hexarotor has a *star-shape*, characterized by the fact that all the O_{P_i}'s are located at the vertexes of a regular hexagon (see Figure 5.4a);

- $\gamma = \frac{\pi}{3}$: the hexarotor has a *Y-shape*, characterized by the fact that the O_{P_i}'s are pairwise located at the vertexes of an equilateral triangle (see Figure 5.4b). To make this configuration practically feasible there must be a suitable vertical distance between each pair of coincident propellers. However, this fact does not change the outcome of the following analysis, and therefore is neglected for the sake of simplicity.

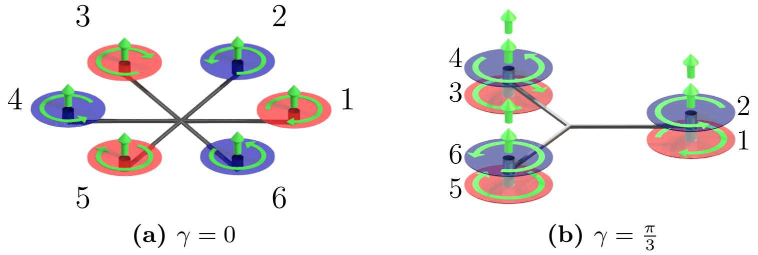


Figure 5.4. Two most popular $(0, 0, \gamma)$-hexarotor GTMs: (a) standard star-shaped hexarotor, and (b) Y-shaped hexarotor.

The angles α and β influence instead only the orientation of the propellers:

- if both $\alpha = 0$ and $\beta = 0$ then the $\mathbf{z}_{P_i}$'s are all pointing in the same direction as $\mathbf{z}_B$. This is the most common situation for standard hexarotors because it is the most efficient in terms of energy. However it results in an under actuated dynamics due to the fact that $\mathrm{rk}(\mathbf{F}) = 1$;
- if $\alpha \neq 0$ and $\beta = 0$ then the $\mathbf{z}_{P_i}$'s are tilted alternatively by an angle α and $-\alpha$ about the axes $\overrightarrow{O_B O_{P_1}} \ldots \overrightarrow{O_B O_{P_6}}$. This choice results in configurations that are less energy-efficient than the previous case. However, their advantage is that one can obtain $\mathrm{rk}(\mathbf{W}) = 6$ which makes the GTM fully actuated.
- if $\alpha = 0$ and $\beta \neq 0$ then the $\mathbf{z}_{P_i}$'s are tilted by an angle β about the axes passing through the O_{P_i}'s and tangential to the circle passing through all the $O_{P_1} \ldots O_{P_6}$. This choice has the same full-actuation pros and energy efficiency cons of the previous case.
- finally, the case in which $\alpha \neq 0$ and $\beta \neq 0$ is a combination of the previous two.

The rest of the section is devoted to the analysis of the role of the angular parameters α, β, γ with respect to the rotor-failure robustness. Specifically, the conditions on these angles are studied which make it possible to realize static hovering after a rotor loss. To this end, $\mathbf{G}(\alpha, \beta, \gamma), \mathbf{H}(\alpha, \beta, \gamma) \in \mathbb{R}^{3\times 6}$ denote the control force and moment input matrices of an (α, β, γ)-hexarotor (i.e., the $\mathbf{F}$ and $\mathbf{M}$ appearing in (5.8), respectively). In addition, ${}^k\mathbf{G}(\alpha, \beta, \gamma)$ and ${}^k\mathbf{H}(\alpha, \beta, \gamma)$ indicate the matrices obtained from $\mathbf{G}(\alpha, \beta, \gamma)$ and $\mathbf{H}(\alpha, \beta, \gamma)$, respectively, by removing the k-th column, i.e., assuming that the k-th propeller fails, with $k \in \mathcal{P}$. Finally, for sake of compactness, the propeller aerodynamic and geometric features are summarized using $r = (c_f/c_\tau)l$, while the symbols s and c stand for sine and cosine, respectively.

The formal results derived in the following are summed up in Table 5.2 that states the influence of the (α, β, γ) angles on full-actuation and rotor-failure robustness properties.

5.4.2 On the Vulnerability of the (0,0,0)-hexarotor GTMs

Before proceeding to analyze the role of the single angular parameters, the case $\alpha = \beta = \gamma = 0$ is considered. This coincides with a standard star-shaped hexarotor.

	role of α	role of β	role of γ
full-actuation	influential	influential	**un**influential
failure full robustness	**un**influential	influential	influential

Table 5.2. A table recalling the role of the angular parameters α, β e γ with respect to the hexarotor actuation properties.

Although highly used, and often believed to be robust to failures, supposedly thanks to the presence of two additional rotors with respect to a quadrotor, these GTMs are actually *fully vulnerable* as stated in the next proposition.

Proposition 5.4.4. *Assume that* $\alpha = \beta = \gamma = 0$*, then the resulting* $(0, 0, 0)$*-hexarotor GTM is fully vulnerable.*

Proof. In Giribet et al. (2016) it has been shown that condition (5.28) is never fulfilled after the loss of any propeller. Therefore Definition 5.3.5 cannot be met as well. □

Here, a new geometrical interpretation of this counterintuitive result is provided which will help both to understand the result itself and to highlight the main drawback of the $(0, 0, 0)$-hexarotor design that should be overcome to attain the robustness against failure of such platforms.

Exploiting (5.34) and imposing $\alpha = \beta = \gamma = 0$, the control moment input matrix of the $(0, 0, 0)$-hexarotor GTM results as in (5.37)

$$\mathbf{H}(0,0,0) = c_\tau \begin{bmatrix} 0 & \frac{\sqrt{3}}{2}r & \frac{\sqrt{3}}{2}r & 0 & -\frac{\sqrt{3}}{2}r & -\frac{\sqrt{3}}{2}r \\ -r & -\frac{1}{2}r & \frac{1}{2}r & r & \frac{1}{2}r & -\frac{1}{2}r \\ 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \tag{5.37}$$

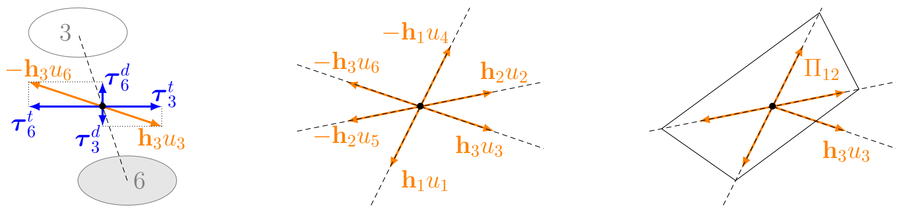
Note that the columns $\mathbf{h}_i \in \mathbb{R}^3$ $i \in \mathcal{P}$ of $\mathbf{H}(0,0,0)$ are such that $\mathbf{h}_1 = -\mathbf{h}_4$, $\mathbf{h}_2 = -\mathbf{h}_5$, $\mathbf{h}_3 = -\mathbf{h}_6$. This means that the total moments generated by the two propellers of an opposed-propeller pair are always collinear regardless of the values assigned to their inputs u_i and u_j, where $(i, j) \in \{(1, 4), (2, 5), (3, 6)\}$ (see Figure 5.5a). It holds that

$$\boldsymbol{\tau}_c = \mathbf{h}_1(u_1 - u_4) + \mathbf{h}_2(u_2 - u_5) + \mathbf{h}_3(u_3 - u_6). \tag{5.38}$$

According to (5.38), the total control moment applied to the platform can be expressed as the linear combination of the linearly independent vectors $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3$ that identify the directions of the moments of opposed-rotor pairs. Given that, even if $u_i, u_j \geq 0$, the sign of $(u_i - u_j)$ can be any, $\boldsymbol{\tau}_c$ can have any direction (and intensity) in $\mathbb{R}^3$ (see Figure 5.5b). However, if any propeller fails, e.g., propeller 6, then $u_6 = 0$ and the control moment degrades to

$$\boldsymbol{\tau}_c = \mathbf{h}_1(u_1 - u_4) + \mathbf{h}_2(u_2 - u_5) + \mathbf{h}_3 u_3. \tag{5.39}$$

Given that u_3 in (5.39) must be non-negative, $\boldsymbol{\tau}_c$ is limited in the half space generated by $\mathbf{h}_3$ and by the delimiting plane Π_{12} parallel to $\mathbf{h}_1$ and $\mathbf{h}_2$, as shown in Figure 5.5c. The



(a) Opposed rotors **(b)** All healthy rotors **(c)** Rotor-failure case

Figure 5.5. Composition of the propeller moments for a $(\alpha, 0, 0)$-hexarotor GTM with any $\alpha \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$.

condition (5.27) is therefore not satisfied, because any $\boldsymbol{\tau}$ belonging to the complementary half-space cannot be attained by any choice of $u_1 \ldots u_5 \geq 0$.

Summarizing,

(i) the total moments generated by two propellers that are opposed are collinear; hence,

(ii) the moments generated by two opposed-rotor pairs ($\mathbf{h}_1$, $\mathbf{h}_4$, $\mathbf{h}_2$, and $\mathbf{h}_5$ in Figure 5.5b) lie all on a 2-dimensional plane, even if they are generated by the (conical) combination of four independently controllable moments; as a consequence,

(iii) five propellers alone can only generate half of the whole 3-dimensional space.

If one finds a way to make the four moments at point (ii) non-coplanar, but actually spanning (by conical combination) the whole space $\mathbb{R}^3$, then symmetry would be broken, singularity overcome, and robustness hopefully achieved. A way to obtain this is to design the hexarotor such that the moments of the opposed propeller pairs are not collinear as in the $(0, 0, 0)$-hexarotor case. It will be shown next which one of the angular parameters of the considered family of hexarotors needs to be changed to actually achieve such goal.

5.4.3 Role of α

Despite the influential role of α in guaranteeing the full-actuation of the (α, β, γ)-hexarotor (see e.g., Ryll et al. (2016); Rajappa et al. (2015)), its effect in the robustness achievement is completely marginal, as summarized in the next statement.

Proposition 5.4.5. *Assume that $\beta = \gamma = 0$, then for any $\alpha \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$ the resulting $(\alpha, 0, 0)$–hexarotor GTM is fully vulnerable.*

Proof. The control moment input matrix for the $(\alpha, 0, 0)$-hexarotor results to be

$$\mathbf{H}(\alpha, 0, 0) = c_\tau \begin{bmatrix} 0 & \frac{\sqrt{3}}{2}f(\alpha, r) & \frac{\sqrt{3}}{2}f(\alpha, r) & 0 & -\frac{\sqrt{3}}{2}f(\alpha, r) & -\frac{\sqrt{3}}{2}f(\alpha, r) \\ -f(\alpha, r) & -\frac{1}{2}f(\alpha, r) & \frac{1}{2}f(\alpha, r) & f(\alpha, r) & \frac{1}{2}f(\alpha, r) & -\frac{1}{2}f(\alpha, r) \\ g(\alpha, r) & -g(\alpha, r) & g(\alpha, r) & -g(\alpha, r) & g(\alpha, r) & -g(\alpha, r) \end{bmatrix}, \quad (5.40)$$

where $f(\alpha, r) = s\alpha + rc\alpha$ and $g(\alpha, r) = c\alpha - rs\alpha$ are scalar values.

Both $\mathbf{H}(\alpha,0,0)$ and ${}^k\mathbf{H}(\alpha,0,0)$ (for any $k\in\mathcal{P}$) are full rank for every value of α in the domain of interest, except when $\tan(\alpha)=-r$ and $\tan(\alpha)=1/r$. In fact, considering $\bar{\mathbf{H}}(\alpha,0,0)=\mathbf{H}(\alpha,0,0)\mathbf{H}^\top(\alpha,0,0)\in\mathbb{R}^{3\times3}$ and ${}^k\bar{\mathbf{H}}(\alpha,0,0)={}^k\mathbf{H}(\alpha,0,0){}^k\mathbf{H}^\top(\alpha,0,0)\in\mathbb{R}^{3\times3}$, it holds that

$$\det(\bar{\mathbf{H}}(\alpha,0,0)) = 54c_\tau^2\,(s\alpha+rc\alpha)^4\,(c\alpha-rs\alpha)^2, \tag{5.41}$$

$$\det({}^k\bar{\mathbf{H}}(\alpha,0,0)) = 27c_\tau^2\,(s\alpha+rc\alpha)^4\,(c\alpha-rs\alpha)^2. \tag{5.42}$$

Trivially, both (5.41)-(5.42) are equal to zero when $\tan(\alpha)=-r$ and $\tan(\alpha)=1/r$, so in these two cases requirement (5.11) is not satisfied and the $(\alpha,0,0)$-hexarotor GTM cannot hover statically.

In other cases, the attention is focused on the requirement (5.28), analyzing $\ker\left({}^k\mathbf{H}(\alpha,0,0)\right)$. Because of the particular structure of the matrix in (5.40), it can be derived that

$$\ker\left({}^k\mathbf{H}(\alpha,0,0)\right) = \operatorname{span}\left(\mathbf{h}^{\backslash k}_{k+1}+\mathbf{h}^{\backslash k}_{k-2},\ \ \mathbf{h}^{\backslash k}_{k+2}+\mathbf{h}^{\backslash k}_{k-1}\right), \tag{5.43}$$

where $\mathbf{h}^{\backslash k}_i$ is the vector of the canonical basis of $\mathbb{R}^5$ obtained in the following way:

1. first, computing the vector of the canonical basis of $\mathbb{R}^6$ which has a one in the entry $i \bmod 6$ and zeros elsewhere,

2. then, removing the k-th entry from the previous vector (which is a zero entry by construction).

For example, if $k=6$ then $\mathbf{h}^{\backslash 6}_{6+1}=[1\ 0\ 0\ 0\ 0]^\top$ and $\mathbf{h}^{\backslash 6}_{6-2}=[0\ 0\ 0\ 1\ 0]^\top$ and therefore $\mathbf{h}^{\backslash 6}_{6+1}+\mathbf{h}^{\backslash 6}_{6-2}=[1\ 0\ 0\ 1\ 0]^\top$. Additionally it occurs $\mathbf{h}^{\backslash 6}_{6+2}+\mathbf{h}^{\backslash 6}_{6-1}=[0\ 1\ 0\ 0\ 1]^\top$. It is easy to check that the last two vectors are in $\ker\left({}^6\mathbf{H}(\alpha,0,0)\right)$ regardless of the value of α.

This implies that any $\mathbf{u}\in\mathbb{R}^5$ that satisfies ${}^6\mathbf{H}(\alpha,0,0)\mathbf{u}=\mathbf{0}$ has one entry structurally equal to 0 (corresponding to the propeller $k+3 \bmod 6$) and therefore (5.28) cannot be satisfied. This finally means that the failed $(\alpha,0,0)$-hexarotor GTM cannot fly in static hovering, namely it is fully vulnerable according to Definition 5.3.5. □

From a geometrical perspective, with reference to Figure 5.5, tilting the propeller 3 of an angle α about $\overrightarrow{O_BO_{P_3}}$ and the propeller 6 of an angle $-\alpha$ about $\overrightarrow{O_BO_{P_6}}$ does tilt the two moments generated by the two opposite rotors in the same way and therefore keeps them collinear. The same holds for the pairs $(1,4)$ and $(2,5)$. As a consequence the discussion provided in Section 5.4.2 is still valid and the vulnerability of the $(\alpha,0,0)$-hexarotor is confirmed by the geometric intuition.

5.4.4 Role of β

The importance of the angle β with respect to the capability of a star-shaped hexarotor to fly after a rotor loss has been discussed in Giribet et al. (2016). To strength the understanding of this fact, in the following it is analytically and geometrically proved that a $(0, \beta, 0)$-hexarotor GTM is also fully robust according to the stronger properties stated in Definition 5.4.3.

Proposition 5.4.6. *Assume that* $\alpha = \gamma = 0$*, then for any non-zero* $\beta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ *such that* $|\tan(\beta)| \neq \sqrt{3}r$ *and* $c^2\beta \neq \frac{1}{(1-r^2)}$*, the resulting* $(0, \beta, 0)$*-hexarotor GTM is fully robust.*

Proof. When $\alpha = \gamma = 0$, the control moment input matrix is parametrized by the angle β:

$$\mathbf{H}(0, \beta, 0) = c_\tau \begin{bmatrix} s\beta & -\frac{1}{2}p(r,\beta) & -\frac{1}{2}p(r,\beta) & s\beta & -\frac{1}{2}q(r,\beta) & -\frac{1}{2}q(r,\beta) \\ -rc\beta & -\frac{1}{2}t(r,\beta) & \frac{1}{2}t(r,\beta) & rc\beta & -\frac{1}{2}s(r,\beta) & \frac{1}{2}s(r,\beta) \\ c\beta & -c\beta & c\beta & -c\beta & c\beta & -c\beta \end{bmatrix}, \tag{5.44}$$

where $p(r,\beta) = s\beta - r\sqrt{3}c\beta$, $q(r,\beta) = s\beta + r\sqrt{3}c\beta$, $s(r,\beta) = \sqrt{3}s\beta - rc\beta$ and $t(r,\beta) = \sqrt{3}s\beta + rc\beta$ are scalar values.

Introducing the matrices $\bar{\mathbf{H}}(0,\beta,0) = \mathbf{H}(0,\beta,0)\mathbf{H}^\top(0,\beta,0) \in \mathbb{R}^{3\times3}$ and ${}^k\bar{\mathbf{H}}(0,\beta,0) = {}^k\mathbf{H}(0,\beta,0){}^k\mathbf{H}^\top(0,\beta,0) \in \mathbb{R}^{3\times3}$, it can first be observed that

$$\det(\bar{\mathbf{H}}(0,\beta,0)) = 54c_\tau^2 c^2\beta \left(1 + (r^2-1)c^2\beta\right)^2, \tag{5.45}$$

$$\det({}^k\bar{\mathbf{H}}(0,\beta,0)) = 27c_\tau^2 c^2\beta \left(1 + (r^2-1)c^2\beta\right)^2. \tag{5.46}$$

Hence, the full-rankness (5.11) is guaranteed for any $\beta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ even in case of any propeller failure, as long as $c^2\beta \neq \frac{1}{(1-r^2)}$.

Then, proceeding as in Giribet et al. (2016), the null space of the matrix ${}^k\mathbf{G}(0,\beta,0)$ can be analyzed, assuming without loss of generality $k = 6$. A generic vector $\mathbf{u} \in \ker({}^6\mathbf{G}(0,\beta,0))$ satisfies the following set of equations

$$u_1 = \frac{\epsilon+1}{2\epsilon}u_3 - \frac{\epsilon+1}{\epsilon-1} \tag{5.47}$$

$$u_2 = u_3 - \frac{\epsilon+1}{\epsilon-1} \tag{5.48}$$

$$u_4 = \frac{\epsilon+1}{2\epsilon}u_3 + 1 \tag{5.49}$$

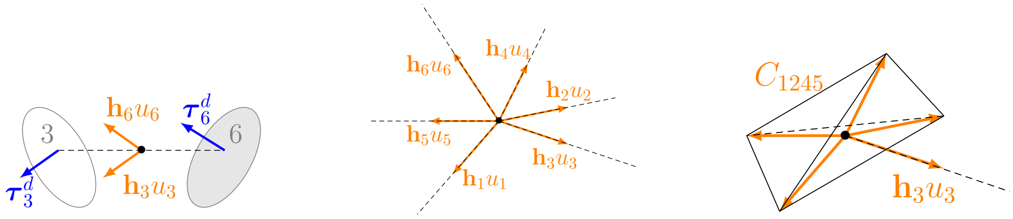
$$u_5 = 1 \tag{5.50}$$

where $\epsilon = -\frac{1}{\sqrt{3}r}\tan(\beta) \in \mathbb{R}$. Hence, supposing that $0 < |\epsilon| < 1$ (or equivalently $0 < |\tan(\beta)| < \sqrt{3}r$), it can be proved that $\mathbf{u} \in \mathbb{R}^5$ defined in (5.47)-(5.50) is strictly positive if $0 < u_3 < \left|\frac{2\epsilon}{\epsilon+1}\right|$. As a consequence, the condition (5.28) is fulfilled.

Using the parametrization (5.47)-(5.50), it can also be proved that ${}^6\mathbf{G}(0,\beta,0)\mathbf{u} \neq \mathbf{0}$, where ${}^6\mathbf{G}(0,\beta,0)$ is obtained removing the 6-th column of the force input matrix

$$\mathbf{G}(0,\beta,0) = \begin{bmatrix} s\beta & \frac{1}{2}s\beta & -\frac{1}{2}s\beta & -s\beta & -\frac{1}{2}s\beta & \frac{1}{2}s\beta \\ 0 & \frac{\sqrt{3}}{2}s\beta & \frac{\sqrt{3}}{2}s\beta & 0 & -\frac{\sqrt{3}}{2}s\beta & -\frac{\sqrt{3}}{2}s\beta \\ c\beta & c\beta & c\beta & c\beta & c\beta & c\beta \end{bmatrix}. \tag{5.51}$$

Because conditions (5.11), (5.28), and (5.30) are met for any $\beta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ such that $|\tan\beta| \neq \sqrt{3}r$ and $c^2\beta \neq \frac{1}{(1-r^2)}$, then the statement of the proposition is proved. □



(a) Opposed rotors

(b) All healthy rotors

(c) Rotor-failure case

Figure 5.6. Composition of the propeller moments for a $(0, \beta, 0)$-hexarotor GTM with any $\beta \neq 0$.

This result can be also justified from the geometric point of view. In fact, when all the propellers are equally inward/outward tilted of an angle $\beta \neq 0$, the total moments of the opposed rotors are not collinear anymore. This is shown in Figure 5.6a for propellers 3 and 6, where the vectors τ_3^d and τ_6^d are rotated in a way that breaks the symmetry while τ_3^t and τ_6^t have the same orientation as in Figure 5.5a so are not shown. Hence, the moments of the opposed propellers, $\mathbf{h}_3 u_3$ and $\mathbf{h}_6 u_6$ in Figure 5.6a, are not collinear anymore and the same holds for the other two pairs of opposed propellers. The total moment is thus the conical combination of six different directions, namely

$$\boldsymbol{\tau}_c = \mathbf{h}_1 u_1 + \mathbf{h}_2 u_2 + \mathbf{h}_3 u_3 + \mathbf{h}_4 u_4 + \mathbf{h}_5 u_5 + \mathbf{h}_6 u_6. \tag{5.52}$$

In this case, the failure of the 6-th propeller does not reduce the total control moment space since, even considering only four of the remaining vectors $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_4, \mathbf{h}_5$ they are not anymore coplanar but actually their conical combination C_{1245} spans the whole $\mathbb{R}^3$ as depicted in Figure 5.6c. The same holds for the failure of any other propeller.

5.4.5 Role of γ

To conclude, the role of γ is evaluated. Note that the condition $\alpha = \beta = 0$ and $\gamma \neq 0$ entails that the propellers are parallel oriented but not equally spaced. This asymmetry of the platform allows to overcome the vulnerability established in Section 5.4.2.

Proposition 5.4.7. *Assume that $\alpha = \beta = 0$, then for any $\gamma \in (0, \frac{\pi}{3}]$, the resulting $(0, 0, \gamma)$-hexarotor GTM is fully robust.*

Proof. Imposing $\alpha = \beta = 0$, the control moment input matrix results to be

$$\mathbf{H}(0,0,\gamma) = c_\tau \begin{bmatrix} 0 & +r\ s\left(\frac{\pi}{3} - \gamma\right) & +r\ \frac{\sqrt{3}}{2} & +r\ s\left(\pi - \gamma\right) & -r\ \frac{\sqrt{3}}{2} & +r\ s\left(\frac{5\pi}{3} - \gamma\right) \\ -r & -r\ c\left(\frac{\pi}{3} - \gamma\right) & +r\ \frac{1}{2} & -r\ c\left(\pi - \gamma\right) & +r\ \frac{1}{2} & -r\ c\left(\frac{5\pi}{3} - \gamma\right) \\ 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \tag{5.53}$$

This is full rank for any choice of $\gamma \in [0, \frac{\pi}{3}]$, and analogously is the derived ${}^k\mathbf{H}(0, 0, \gamma)$ for any $k \in \mathcal{P}$. This fact can be verified by considering the determinant of the matrices

$\bar{\mathbf{H}}_2(0,0,\gamma) = \mathbf{H}(0,0,\gamma)\mathbf{H}^\top(0,0,\gamma) \in \mathbb{R}^{3\times3}$ and ${}^k\bar{\mathbf{H}}_2(0,0,\gamma) =^k \mathbf{H}(0,0,\gamma)^k\mathbf{H}^\top(0,0,\gamma) \in \mathbb{R}^{3\times3}$. Specifically, it occurs that $\det(\bar{\mathbf{H}}_2(0,0,\gamma)) = 54c_\tau^2 r^4$, hence the condition (5.11) is always fulfilled independently from γ. In case of any rotor failure, the determinant of ${}^k\bar{\mathbf{H}}_2(0,0,\gamma)$ results instead to be a complex non-linear function of γ, however it can be numerically checked that it is never null in the domain of interest. Hence, the first condition for the static hovering realizability is always satisfied in case of rotor-failure.

To explore which conditions on γ possibly ensure that ${}^k\mathbf{H}(0,0,\gamma)$ fulfills requirement (5.28) it is assumed again that the 6-th rotor fails. The solution of ${}^6\mathbf{H}(0,0,\gamma)\mathbf{u} = \mathbf{0}$ can then be written in the following form

$$u_1 = u_4 + \frac{(-\sqrt{3}s\gamma - c\gamma + 1)}{(2c\gamma + 1)} \tag{5.54}$$

$$u_2 = -\frac{\left(\sqrt{3}s\gamma - c\gamma + 1\right)}{(2c\gamma + 1)}u_4 + \frac{3}{(2c\gamma + 1)} \tag{5.55}$$

$$u_3 = -\frac{\left(\sqrt{3}s\gamma - c\gamma + 1\right)}{(2c\gamma + 1)}u_4 + \frac{\left(\sqrt{3}s\gamma - c\gamma + 1\right)}{(2c\gamma + 1)} \tag{5.56}$$

$$u_5 = 1 \tag{5.57}$$

Observing that $0 \leq s\gamma \leq \frac{\sqrt{3}}{2}$ and $\frac{1}{2} \leq c\gamma \leq 1$ in the domain of interest, it can be verified that the positivity of $\mathbf{u}$ is ensured only if $\gamma > 0$. In other words, the condition $\gamma > 0$ implies the existence of a strictly positive vector $\mathbf{u} \in \ker({}^6\mathbf{H}(0,0,\gamma))$, namely the fulfillment of (5.28).

Exploiting (5.54)-(5.57), it is possible to show that also requirement (5.30) is satisfied when $\gamma > 0$. To do so, it is necessary to evaluate the relation ${}^6\mathbf{G}(0,0,\gamma)\mathbf{u}$ by introducing the control force input matrix

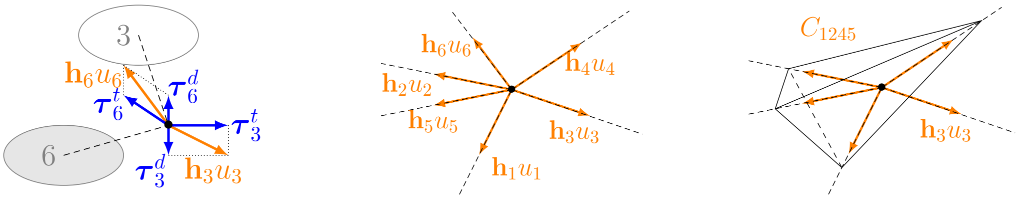
$$\mathbf{G}(0,0,\gamma) = c_f \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{5.58}$$

Trivially, it results that ${}^6\mathbf{G}(0,0,\gamma)\mathbf{u} \neq \mathbf{0}$. As a consequence, both the Y-shape hexarotor ($\gamma = \frac{\pi}{3}$) and all the less common configurations where $0 < \gamma < \frac{\pi}{3}$ are fully robust. □

Figure 5.7a shows the moments composition for a pair of opposed rotors in a Y-shaped hexarotor. It is straightforward that whenever $\gamma > 0$ the moment directions of the opposed propellers are not collinear anymore. This generates the same beneficial consequences described in Section 5.4.4 as shown in Figures 5.7b and 5.7c. In particular, for example, the conical combination C_{1245} spans the whole $\mathbb{R}^3$ also in this case.

5.5 Results Summary

This chapter is devoted to the introduction (and the preliminary analysis) of the multi-rotor UAVs, that can be considered as agents composing a complex multi-vehicle formation



(a) Opposed rotors

(b) All healthy rotors

(c) Rotor-failure case

Figure 5.7. Composition of the propeller moments for a $(0, 0, \gamma)$-hexarotor GTM with $\gamma = \frac{\pi}{3}$.

(see Chapter 7)[2]. The autonomous aerial platforms constitutes an emerging technology whose exploitation within smart environments context is rapidly growing also thanks to the fervent interest of the robotics community in design, modeling and control of such vehicles.

In this perspective, in this chapter two fundamental actuation properties have been studied for the platforms classified as generically-tilted multi-rotors.

First, the interplay between the control force and the control moment has been considered. It has been distinguished between fully coupled, partially coupled and fully decoupled platforms according to both the dimension of the freely assignable force space $\mathfrak{F}_B$ and its relation with the total force space $\mathfrak{F}$. Then, it has been introduced the concept of static hovering realizability which rests upon the possibility to reject any disturbance torque while counterbalancing the gravity.

The robustness properties of a family of hexarotors parametrized by three angles are finally explored in terms of capability to statically hover after a rotor loss. It has turned out that the full robustness is guaranteed by (inward/outward) tilting each propeller on its $\overrightarrow{O_B O_{P_i}}$-axis or by moving towards the Y-shaped hexarotor and thus breaking the symmetry of the propeller positions in the star-shaped hexarotor.

[2] Alternatively, UAVs can be regarded as multi-agent systems them shelves composed by a set of propellers.

6

Static Hovering Control of a UAV

Nowadays multi-rotor aerial vehicles are employed in several application fields ranging from the classical visual sensing tasks (e.g., surveillance and aerial photography) to the modern environment exploration and physical interaction (e.g., search and rescue operations, grasping and manipulation). In these frameworks, static hovering constitute a classical control issue and a suitable flight condition in case of a motor failure potentially allowing the damage platform to slowly and gently land on the ground.

In this chapter, two controllers are illustrated aiming at keeping a given multi-rotor UAV platform in a constant spot without rotating. The first control law faces the non-linearity of the vehicle motion equations through a state feedback structure, while the second solution exploits the intrinsic cascaded nature of the multi-rotor dynamics.

The contents of this chapter are available in
Michieletto G., Cenedese A., Zaccarian L., and Franchi A. Nonlinear control of multi-rotor aerial vehicles based on the zero-moment direction. *IFAC World Congress 2017*, pages 13686–13691, 2017a;
Michieletto G., Ryll M., and Franchi A. Control of statically hoverable multi-rotor aerial vehicles and application to rotor-failure robustness for hexarotors. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2747–2752, 2017b.

6.1 Static Hovering Control Task

A generically-tilted multi-rotor platform (GTM platform introduced in Section 5.1.2) can hover statically when it is able to reject torque disturbances in any direction while counterbalancing the gravity. According to Definition 5.3.5, the first requirement is guaranteed by the fulfilment of conditions (5.11) and (5.28), whereas condition (5.30) allows to realize a control force with any intensity. When all these conditions are met, there exists a control law that assigns the multi-rotor input commands to steer and keep it in a fixed reference position stabilizing the attitude.

The design of the suitable controller for a single vehicle represents the first step towards the control of a flying multi-agent formation, wherein the dynamics of each platform is required to be locally governed to achieve global tasks. Specifically, statically hovering represents a beneficial flight condition for the fulfillment of surveillance goals that implies the need of multiple agents (e.g. because of the size of the environment of interest). In this optics, this chapter constitutes a necessary introduction for the contents of Chapter 7.

6.1.1 Literature Overview

Statically hovering flight condition results to be useful in many surveillance applications, but also in case of propellers loss. In literature many control strategies are known to enhance the stability of a UAV able to achieve this task. These are usually based on linear control systems such as proportional-derivative controllers or linear quadratic regulators, see e.g., Erginer and Altug (2007); Argentim, Rezende, Santos, and Aguiar (2013). Non-linear controllers are not equally popular and mainly exploit feedback linearization (Mistler, Benallegue, and M'sirdi (2001)), sliding mode and backstepping (Bouabdallah and Siegwart (2005)) and geometric control (Lee, Kim, and Sastry (2009)).

Although less diffused comparing to the linear solutions, the effectiveness of non-linear control techniques has been widely confirmed by the experimental tests. For example, Carrillo, Dzul, and Lozano (2012) have experimentally evaluated the performance of three different non-linear strategies for stabilizing a quadrotor. Specifically, they have analyzed the performance of controllers based on nested saturations, backstepping and sliding mode in stabilizing the position of the vehicle with respect to an artificial visual landmark on the ground. Similarly, Choi and Ahn (2015) have validated the possibility for a real quadrotor to stably track a point by using a non-linear control strategy that rests upon a backstepping-like feedback linearization method.

It can be observed that the non-linear strategies have been used mainly for under-actuated GTMs, often exploiting their cascaded dependent dynamic. The most popular solution in this sense has been proposed in Lee, Leoky, and McClamroch (2010), where a geometric trajectory tracking controller for a quadrotor UAV is introduced. In particular, through the Lyapunov theory, it has been shown that this controller, developed in the three-dimensional Special Euclidean space $SE(3)$, exhibits almost global exponential attractiveness to the zero equilibrium of tracking errors. However, cascaded geometric control solutions can be used as well for fully-actuated GTMs in order to, e.g., handle actuator saturations. This is the case evaluated in Franchi et al. (2016), where a new lower-level cascaded full-pose controller has been introduced which accepts as reference a pose trajectory and modifies it (prioritizing the position tracking) when needed by the limitation imposed by spinning rate saturations of the propellers.

6.1.2 Problem Formalization

Recalling Definition 5.3.1, a given GTM platform is in equilibrium when its linear and angular velocity (and accelerations) are zeros. Hence, static hovering represents an equilibrium condition wherein the pose of the vehicle is constant over the time.

Given a multi-rotor vehicle whose dynamics is governed by system (5.9)-(5.10), the static hovering control deals with the selection of the command input vector $\mathbf{u}$ so that for any constant reference position $\mathbf{p}_r \in \mathbb{R}^3$, the closed-loop system is able to asymptotically stabilize the position $\mathbf{p}$ of the GTM at $\mathbf{p}_r$ with some hovering orientation. In other words, the controller asymptotically stabilizes a set where $\mathbf{p} = \mathbf{p}_r$, and $\dot{\mathbf{p}}$ and $\boldsymbol{\omega}$ are both zero, while the orientation $\mathbf{R}$ could be arbitrary but constant. An additional requirement can then be imposed on the attitude, seeking that this is as close as possible to a given constant reference orientation $\mathbf{R}_r \in SO(3)$.

In the rest of the chapter two control solutions to steer and keep a given GTM platform in such a static hovering condition are presented. Both the designed controllers rely on the existence of at least a preferential direction $\mathbf{d}_* \in \mathfrak{F}_B$ in the force space that can be generically oriented. A good and viable choice is to set $\mathbf{d}_*$ as in Proposition 5.3.9, namely

$$\mathbf{d}_* = \frac{\mathbf{F}\bar{\mathbf{u}}}{\|\mathbf{F}\bar{\mathbf{u}}\|_2} = \frac{\mathbf{F}\bar{\mathbf{u}}}{\xi}, \tag{6.1}$$

with $\bar{\mathbf{u}}$ satisfying condition (5.28) and $\mathbf{F}$ being the control force input matrix (5.8). Note that in state-of-the-art multi-rotor controllers it is implicitly assumed that $\mathbf{d}_*$ is oriented along the z-axis of the vehicle body frame and that the considered GTM is fully decoupled. Contrarily, in both the proposed controllers, $\mathbf{d}_*$ can be any and the platform can be also partially coupled.

The first control solution exploits a sort of dynamic feedback linearization around the preferential direction. In addition, the quaternion formalism is used to represent the platform attitude (the corresponding motion equations are provided in Section 6.2.1), in order to simplify the computations needed to formally prove its convergence.

The second control solution, instead, presents a cascaded structure. The idea is to implement, independently from the position dynamics, an inner control loop that achieves the feedback control of the angular velocity $\boldsymbol{\omega}$ and of the attitude $\mathbf{R}$ exploiting the moment control input $\boldsymbol{\tau}_c$ that appears in (5.10). Then the orientation $\mathbf{R}$ is considered as an additional control input and uses in (5.9), with the control force $\mathbf{f}_c$, to implement an outer loop for the feedback control of the position $\mathbf{p}$ and the linear velocity $\dot{\mathbf{p}}$.

6.2 Quaternion-Based State Feedback Controller

This section is devoted to the description of the first control solution for a GTM platform that is required to fly in static hovering. The proposed controller exploits a state feedback structure to cope with the non-linear nature of the dynamic equations and is designed by using the quaternion formalism (in place of the rotation matrices) to represent the vehicle attitude.

Beyond the conditions in Definition 5.3.5, the platform is assumed to satisfy also the following requirement on the control force and moment input matrices that determine its dynamics, i.e., on $\mathbf{F}$ and $\mathbf{M}$ in (5.8), respectively. In detail, it is taken for granted that

$$\exists \mathbf{K} \in \mathbb{R}^{n \times n} \quad \text{such that} \quad \mathbf{F}\mathbf{K}\mathbf{M}^\top(\mathbf{M}\mathbf{K}\mathbf{M}^\top)^{-1} = \mathbf{F}\,\mathbf{M}_K^\dagger = \mathbf{0}. \tag{6.2}$$

This hypothesis, where $\mathbf{M}_K^\dagger \in \mathbb{R}^{n \times 3}$ denotes the *generalized right pseudo-inverse* of the control moment input matrix, translates into the requirement that the row space of $\mathbf{F}$ is orthogonal to the row space of $\mathbf{M}$ transformed via matrix $\mathbf{K}$. This constraint essentially enables a sufficient level of decoupling between the control force $\mathbf{f}_c$ and the control moment $\boldsymbol{\tau}_c$, and is not too restrictive. For example, for the consider classical multi-rotor systems introduced in Section 5.2.1 the condition $\mathbf{F}\mathbf{M}_K^\dagger = \mathbf{0}$ is satisfied by simply choosing $\mathbf{K} = \mathbf{I}_n$. This configuration is the one considered in Lee et al. (2010), nevertheless, condition (6.2) is much weaker, as clearly illustrated by the example study of Section 6.2.6.

6.2.1 Quaternion-Based Attitude Representation

Given an aerial platform, the orientation of the body frame $\mathscr{F}_B$ with respect to the world frame $\mathscr{F}_W$ can always be represented by the unit quaternion $\mathbf{q} = [\eta \;\; \boldsymbol{\epsilon}^\top]^\top \in \mathbb{S}^3$, in place of the the rotation matrix $\mathbf{R} \in SO(3)$. The main notions on the quaternion mathematics are given in Appendix B, however in the following the consequences of the use of this attitude representation in the GTM kinematic and dynamic model are evaluated.

The principal difference and advantage in the use of the quaternion convention is the fact that the orientation kinematics of the GTM vehicle is given by a *linear relation*, contrarily to the rotation matrices case (see (5.1)). Specifically, when the angular velocity $\boldsymbol{\omega}$ of $\mathscr{F}_B$ is expressed in body frame, it holds that

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \odot \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = \frac{1}{2}\mathbf{V}(\mathbf{q}) \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}, \tag{6.3}$$

where it has been exploited the standard quaternions composition rule (B.18), using

$$\mathbf{V}(\mathbf{q}) = \begin{bmatrix} \eta & -\boldsymbol{\epsilon}^\top \\ \boldsymbol{\epsilon} & \eta\mathbf{I}_3 + [\boldsymbol{\epsilon}]_\times \end{bmatrix} \in \mathbb{R}^{4 \times 4}. \tag{6.4}$$

On the other hand, as each unit quaternion $\mathbf{q}$ is univocally associated to a rotation matrix via $\mathbf{R}(\mathbf{q}) = \mathbf{I}_3 + 2\eta[\boldsymbol{\epsilon}]_\times + 2\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top - 2\boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}\mathbf{I}_3$, the translational kinematic of a GTM is governed by $\dot{\mathbf{p}} = \mathbf{R}(\mathbf{q})\mathbf{v}$. Note that the same equation is valid when the rotation matrix formalism is used discarding the dependency of $\mathbf{R}$ on the corresponding quaternion.

Finally, the dynamics of the platform is dictated by the following relations whose physical meaning is unchanged comparing to (5.9)-(5.10), namely

$$m\ddot{\mathbf{p}} = -mg\mathbf{e}_3 + \mathbf{R}(\mathbf{q})\mathbf{F}\mathbf{u} \tag{6.5}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}\mathbf{u} \tag{6.6}$$

where m and $\mathbf{J}$ are the inertial parameters and g is the gravity constant.

6.2.2 Controller Scheme

One of the main elements of the designed controller (Figure 6.1) is the identification of a zero-moment preferential direction $\mathbf{d}^*$ in the force space of the considered GTM platform. As stated in Section 6.1.2, according to Proposition 5.3.9, this is assumed to be

$$\mathbf{d}_* = \frac{\mathbf{F}\bar{\mathbf{u}}}{\|\mathbf{F}\bar{\mathbf{u}}\|_2} = \frac{\mathbf{F}\bar{\mathbf{u}}}{\xi}, \tag{6.7}$$

where $\bar{\mathbf{u}} \in \mathbb{R}^n$ satisfies condition (5.30) for the static hoverability, so that $\bar{\mathbf{u}} \in \ker(\mathbf{M})$.

Under the additional requirement (6.2) about the existence of the generalized pseudo-inverse of the moment control input matrix $\mathbf{M}$, the dynamic controller here described yields a command input vector $\mathbf{u}$ having the following expression

$$\mathbf{u} = \mathbf{M}_K^\dagger \boldsymbol{\tau}_r + \bar{\mathbf{u}} f, \tag{6.8}$$

so that vector $\boldsymbol{\tau}_r \in \mathbb{R}^3$ and scalar $f \in \mathbb{R}$ appear conveniently in the expression of the control force and the control moment implying

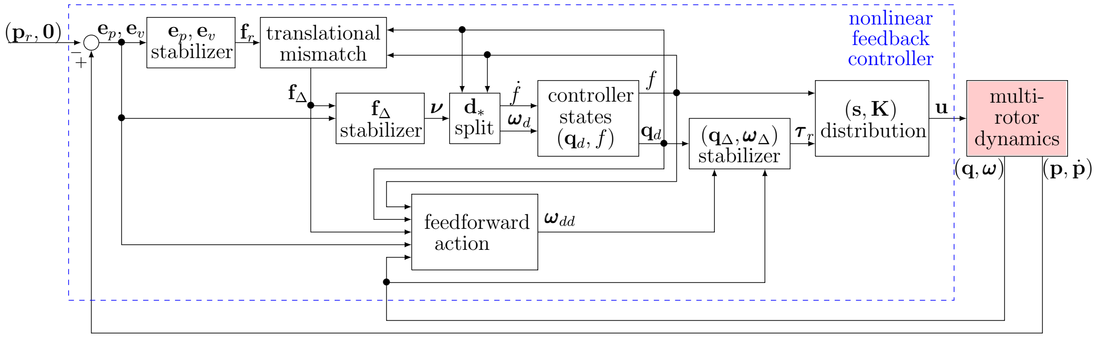
$$\mathbf{f}_c = \mathbf{F}\mathbf{u} = \mathbf{d}_* \xi f, \tag{6.9}$$

$$\boldsymbol{\tau}_c = \mathbf{M}\mathbf{u} = \boldsymbol{\tau}_r, \tag{6.10}$$

which clearly reveals a nice decoupling feature behind (6.2) and selection (6.8). Once this decoupling is in place, it is of interest to steer the platform in a *desired orientation* $\mathbf{q}_d \in \mathbb{S}^3$ such that the direction of the resulting force $\mathbf{R}(\mathbf{q}_d)\mathbf{f}_c$ acting on the translational dynamics (6.5) (i.e., the direction of $\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*$) coincides with a desired direction arising from a simple PD + gravity compensation feedback function. This is here selected as

$$\mathbf{f}_r = mg\mathbf{e}_3 - k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v, \tag{6.11}$$

where $\mathbf{e}_p = \mathbf{p} - \mathbf{p}_r \in \mathbb{R}^3$ and $\mathbf{e}_v = \dot{\mathbf{p}} \in \mathbb{R}^3$ are the *position error* and the *velocity error* (in world frame), respectively, while $k_{pp} \in \mathbb{R}^+$ and $k_{pd} \in \mathbb{R}^+$ are arbitrary PD gains.



$(\mathbf{p}_r, \mathbf{0})$
$\mathbf{e}_p, \mathbf{e}_v$
$-$
$+$
$\mathbf{e}_p, \mathbf{e}_v$ stabilizer
$\mathbf{f}_r$
translational mismatch
$\mathbf{f}_\Delta$
$\mathbf{f}_\Delta$ stabilizer
$\boldsymbol{\nu}$
$\mathbf{d}_*$ split
$\dot{f}$
$\boldsymbol{\omega}_d$
controller states $(\mathbf{q}_d, f)$
f
$\mathbf{q}_d$
$(\mathbf{q}_\Delta, \boldsymbol{\omega}_\Delta)$ stabilizer
$\boldsymbol{\tau}_r$
$(\mathbf{s}, \mathbf{K})$ distribution
$\mathbf{u}$
nonlinear feedback controller
multi-rotor dynamics
$(\mathbf{q}, \boldsymbol{\omega})$
$(\mathbf{p}, \dot{\mathbf{p}})$
feedforward action
$\boldsymbol{\omega}_{dd}$

Figure 6.1. Block diagram of the closed-loop system with the proposed feedback control strategy.

Rather than directly choosing $\mathbf{q}_d$, it is preferable to introduce an auxiliary state in the controller, evolving in $\mathbb{S}^3$ through the quaternion-based dynamics in (6.3), namely

$$\dot{\mathbf{q}}_d = \frac{1}{2}\mathbf{q}_d \odot \begin{bmatrix} 0 \\ \boldsymbol{\omega}_d \end{bmatrix}, \tag{6.12}$$

where $\boldsymbol{\omega}_d \in \mathbb{R}^3$ is an additional virtual input that should be selected so that the actual input to the translational dynamics eventually converges to the state feedback in (6.11). In other words, $\boldsymbol{\omega}_d$ should be set in such a way to drive to zero the following mismatch, motivated by (6.5) and (6.9),

$$\mathbf{f}_\Delta = \mathbf{R}(\mathbf{q}_d)\mathbf{f}_c - \mathbf{f}_r \tag{6.13}$$

$$= \mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\xi f - \mathbf{f}_r. \tag{6.14}$$

In particular, this is ensured by considering the variable f in (6.8) as an additional scalar state of the controller, and then imposing

$$\boldsymbol{\omega}_d = \frac{1}{\xi f}\left[\mathbf{d}_*\right]_\times \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu}, \tag{6.15}$$

$$\dot{f} = \frac{1}{\xi}\left(\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\right)^\top \boldsymbol{\nu}, \tag{6.16}$$

where it is

$$\boldsymbol{\nu} = \left(\frac{k_{pd}k_{pp}}{m}\mathbf{e}_p + \left(\frac{k_{pd}^2}{m} - k_{pp}\right)\mathbf{e}_v - \left(\frac{k_{pd}}{m} + k_\Delta\right)\mathbf{f}_\Delta\right), \tag{6.17}$$

being k_Δ an additional positive scalar gain, i.e., $k_\Delta \in \mathbb{R}^+$. Note that the equation (6.15) clearly makes sense only if $f \neq 0$ (this is guaranteed by the stated assumptions on the platform - in particular it is formally established in Proposition 6.2.1 in Section 6.2.4).

The scheme is completed by an appropriate selection of $\boldsymbol{\tau}_r$ in (6.8) ensuring that the actual attitude $\mathbf{q}$ indeed tracks the desired attitude $\mathbf{q}_d$. This task is simple because of the additional requirement (6.2), which guarantees the full-authority control action on the rotational dynamics.

To simplify the exposition, let introduce the mismatch $\mathbf{q}_\Delta \in \mathbb{S}^3$ between the current and the desired orientation, namely

$$\mathbf{q}_\Delta = \mathbf{q}_d^{-1} \odot \mathbf{q} = \begin{bmatrix} \eta_d \eta + \boldsymbol{\epsilon}_d^\top \boldsymbol{\epsilon} \\ -\eta \boldsymbol{\epsilon}_d + \eta_d \boldsymbol{\epsilon} - [\boldsymbol{\epsilon}_d]_\times \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \eta_\Delta \\ \boldsymbol{\epsilon}_\Delta \end{bmatrix}. \tag{6.18}$$

Then the *reference moment* $\boldsymbol{\tau}_r$ in (6.8) ensuring the convergence to zero of this mismatch is selected as

$$\boldsymbol{\tau}_r = -k_{ap}\boldsymbol{\epsilon}_\Delta - k_{ad}\boldsymbol{\omega}_\Delta + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{J}\boldsymbol{\omega}_{dd}, \tag{6.19}$$

where $\boldsymbol{\omega}_\Delta = \boldsymbol{\omega} - \boldsymbol{\omega}_d \in \mathbb{R}^3$ is the angular velocity mismatch and the PD gains $k_{ap} \in \mathbb{R}^+$ and $k_{ad} \in \mathbb{R}^+$ allow to tune the proportional and derivative action of the attitude transient.

In (6.19), one clearly sees a feedforward term cancelling out the quadratic term in $\boldsymbol{\omega}$ appearing in (6.6), in addition to a correction term $\boldsymbol{\omega}_{dd}$ ensuring the forward invariance of the set where $\mathbf{q} = \mathbf{q}_d$ and $\boldsymbol{\omega} = \boldsymbol{\omega}_d$. For this purpose, $\boldsymbol{\omega}_{dd}$ is chosen as

$$\begin{aligned} \boldsymbol{\omega}_{dd} = \frac{1}{\xi f}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\,(k_1 \mathbf{R}(\mathbf{q})\mathbf{d}_* \xi f + k_2(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\mathbf{e}_p + \\ k_3(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\mathbf{e}_v + k_4(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\mathbf{f}_\Delta)\,, \end{aligned} \tag{6.20}$$

where

$$k_1 = \frac{k_{pd}^2}{m^2} - \frac{k_{pp}}{m} \tag{6.21}$$

$$k_2(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta) = -\left(\frac{k_{pd}^2 k_{pp}}{m^2} + \frac{k_{pp}^2}{m} + \kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\,\frac{k_{pd}k_{pp}}{m}\right) \tag{6.22}$$

$$k_3(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta) = -\left(\frac{k_{pd}^2 k_{pp}}{m^2} + \frac{k_{pp}^2}{m} + \kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\,\frac{k_{pd}k_{pp}}{m}\right) \tag{6.23}$$

$$k_4(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta) = \frac{k_{pd}^2}{m^2} - \frac{k_{pp}}{m} + \frac{k_{pd}k_\Delta}{m} + k_\Delta^2 + \kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\,\left(\frac{k_{pd}}{m} + k_\Delta\right) \tag{6.24}$$

and

$$\kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta) = -\frac{2}{\xi f}\mathbf{d}_*^\top \mathbf{R}^\top(\mathbf{q}_d)\left(\frac{k_{pd}k_{pp}}{m}\mathbf{e}_p + \left(\frac{k_{pd}^2}{m} - k_{pp}\right)\mathbf{e}_v - \left(\frac{k_{pd}}{m} + k_\Delta\right)\mathbf{f}_\Delta\right) \tag{6.25}$$

It can be proved that $\boldsymbol{\omega}_{dd}$ in (6.20) corresponds to the derivative of $\boldsymbol{\omega}_d$ (the proof is available in Appendix A), while observing Figure 6.1, which depicts the structure of the described controller, one can note that the feedforward variable depends on the current orientation and angular velocity of the vehicle.

6.2.3 Error Dynamics

To analyze the properties of the closed-loop system presented in the previous section, the following relevant dynamics are introduced for the *orientation error* variable $\mathbf{q}_\Delta$ in (6.18) and the associated *angular velocity mismatch* $\boldsymbol{\omega}_\Delta$, i.e.,

$$\dot{\mathbf{q}}_\Delta = \frac{1}{2}\mathbf{q}_\Delta \odot \begin{bmatrix} 0 \\ \boldsymbol{\omega}_\Delta \end{bmatrix}, \tag{6.26}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}}_\Delta = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} - \mathbf{J}\dot{\boldsymbol{\omega}}_d + \boldsymbol{\tau}_r. \tag{6.27}$$

To establish useful properties of the translational dynamics, the error vector $\mathbf{e}_t = \begin{bmatrix}\mathbf{e}_p^\top & \mathbf{e}_v^\top\end{bmatrix}^\top \in \mathbb{R}^6$ is introduced, which well characterizes the deviation from the constant reference position $\mathbf{p}_r \in \mathbb{R}^3$. Combining equation (6.5) with the definition of $\mathbf{f}_\Delta$ given in (6.14) the dynamics of $\mathbf{e}_t$ can be rewritten as follows

$$\dot{\mathbf{e}}_p = \mathbf{e}_v \tag{6.28}$$

$$m\dot{\mathbf{e}}_v = -mg\mathbf{e}_3 + (\mathbf{R}(\mathbf{q}) - \mathbf{R}(\mathbf{q}_d))\mathbf{f}_c + \mathbf{f}_r + \mathbf{f}_\Delta. \tag{6.29}$$

A last variable that needs to be characterized is the controller state f. In particular, combining (6.5) with (6.9), it results that the zero position error condition $\mathbf{e}_p = \mathbf{0}$ can only be reached if the state f, governed by (6.16), converges to the value mg/ξ. Rather than describing the error system in terms of the deviation $f - mg/\xi$ (which should clearly go to zero), here it is preferred to use the redundant set of coordinates $\mathbf{f}_\Delta$. Indeed, according to (6.14), showing that $\mathbf{f}_\Delta$ tends to zero implies that, asymptotically, it results $\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\xi f = \mathbf{f}_r$. Namely, as long as $\mathbf{e}_t$ tends to zero, the set where $\mathbf{d}_*\xi f = mg\mathbf{R}^\top(\mathbf{q}_d)\mathbf{e}_3$ is approached. Because $\mathbf{q}_\Delta = \mathbf{q}_I$ implies $\mathbf{R}(\mathbf{q}) = \mathbf{R}(\mathbf{q}_d)$, this clearly corresponds to the set wherein the orientation satisfies $\mathbf{R}(\mathbf{q})\mathbf{d}_* = \mathbf{R}(\mathbf{q}_d)\mathbf{d}_* = \mathbf{e}_3$ and $|f| = mg/\xi$.

6.2.4 Stability Analysis

The error variables, whose closed-loop dynamics has been characterized in the previous section, can be used to prove that the proposed control scheme solves the static hovering control task. To formalize this observation, let consider the following state for the overall closed loop

$$\mathbf{z} = (\mathbf{q}_\Delta, \boldsymbol{\omega}_\Delta, \mathbf{f}_\Delta, \mathbf{e}_t, \mathbf{q}) \in \mathcal{Z}, \tag{6.30}$$

where $\mathcal{Z} = \mathbb{S}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^6 \times \mathbb{S}^3 \subseteq \mathbb{R}^{20}$, and the compact set

$$\mathcal{Z}_0 = \left\{\mathbf{z} \in \mathcal{Z} \mid \mathbf{q}_\Delta = \mathbf{q}_I, \boldsymbol{\omega}_\Delta = \mathbf{0}, \mathbf{f}_\Delta = \mathbf{0}, \mathbf{e}_t = \mathbf{0}, \mathbf{R}(\mathbf{q})\mathbf{d}_* = \mathbf{e}_3\right\}, \tag{6.31}$$

which clearly characterizes the requirement that the desired position is asymptotically reached ($\mathbf{e}_t = \mathbf{0}$), with some constant orientation ensuring that the zero-moment direction $\mathbf{d}_*$ is correctly aligned with the steady-state action $mg\mathbf{e}_3$, necessary for counteracting the gravity force. Note that set $\mathcal{Z}_0$ is compact because it is the product of compact sets.

Before proceeding with the proof, a useful property of the compact set $\mathcal{Z}_0$ is established showing that the controller state f is nonzero.

Proposition 6.2.1. *There exists a neighborhood of the compact set $\mathcal{Z}_0$ where variable f is (uniformly) lower bounded from zero.*

Proof. Since in $\mathcal{Z}_0$ it occurs that $\mathbf{e}_t = \mathbf{0}$ and $\mathbf{f}_\Delta = \mathbf{0}$, then from (6.14) it follows that $\mathbf{d}_* \xi f = mg\mathbf{R}^\top(\mathbf{q}_d)\mathbf{e}_3$. Taking norms on both sides and due to the property of rotation matrices, it holds that $|f| = mg/\xi$, because $\xi = \|\mathbf{F}\bar{\mathbf{u}}\|_2 > 0$. Since $\mathcal{Z}_0$ is compact, by continuity there exists a neighborhood of $\mathcal{Z}_0$ where $|f|$ is (uniformly) positively lower bounded. □

The proof of stability is carried out by focusing on increasingly small nested sets, each of them characterized by a desirable behavior of certain components of the variable $\mathbf{z}$ in (6.30). The first set corresponds to the set where the attitude mismatch $(\mathbf{q}_\Delta, \boldsymbol{\omega}_\Delta)$ is null. It is defined as follows and it is clearly an unbounded and closed set

$$\mathcal{Z}_a = \{\mathbf{z} \in \mathcal{Z} \mid \mathbf{q}_\Delta = \mathbf{q}_I,\ \boldsymbol{\omega}_\Delta = \mathbf{0}\}\,. \tag{6.32}$$

For this set, the following result can be proved, establishing that solutions remaining close to the compact set $\mathcal{Z}_0$ are well behaved in terms of asymptotic stability of the non-compact set $\mathcal{Z}_a$.

Lemma 6.2.2. *Set $\mathcal{Z}_a$ is locally asymptotically stable near $\mathcal{Z}_0$ for the closed-loop dynamics.*

Proof. The thesis can be proved exploiting the dynamics of the variables $\mathbf{q}_\Delta$ and $\boldsymbol{\omega}_\Delta$ in (6.26) and (6.27) and defining the Lyapunov function

$$V_a = 2k_{ap}(1 - \eta_\Delta) + \frac{1}{2}\boldsymbol{\omega}_\Delta^\top \mathbf{J}\boldsymbol{\omega}_\Delta, \tag{6.33}$$

which is clearly positive definite in a neighborhood of $\mathcal{Z}_a$. Using equations (6.19), (6.26), (6.27), which hold close to $\mathcal{Z}_0$ due to the result established in Proposition 6.2.1, the dynamics restricted to variables $\mathbf{q}_\Delta$ and $\boldsymbol{\omega}_\Delta$ results to be

$$\dot{\mathbf{q}}_\Delta = \begin{bmatrix} \dot{\eta}_\Delta \\ \dot{\boldsymbol{\epsilon}}_\Delta \end{bmatrix} = \frac{1}{2}\mathbf{W}(\mathbf{q}_\Delta)\begin{bmatrix} 0 \\ \boldsymbol{\omega}_\Delta \end{bmatrix}, \tag{6.34}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}}_\Delta = -k_{ap}\boldsymbol{\epsilon}_\Delta - k_{ad}\boldsymbol{\omega}_\Delta, \tag{6.35}$$

which is clearly autonomous (it does not depend on external signals). Then, the derivative of V_a along the dynamics turns out to be

$$\begin{aligned} \dot{V}_a &= -2k_{ap}\dot{\eta}_\Delta + \boldsymbol{\omega}_\Delta^\top \mathbf{J}\dot{\boldsymbol{\omega}}_\Delta && (6.36)\\ &= k_{ap}\boldsymbol{\omega}_\Delta^\top \boldsymbol{\epsilon}_\Delta + \boldsymbol{\omega}_\Delta^\top(-k_{ap}\boldsymbol{\epsilon}_\Delta - k_{ad}\boldsymbol{\omega}_\Delta) && (6.37)\\ &= -k_{ad}\|\boldsymbol{\omega}_\Delta\|^2. && (6.38) \end{aligned}$$

Finally, since the dynamics is autonomous, and the set where both $\mathbf{q}_\Delta$ and $\boldsymbol{\omega}_\Delta$ are zero is compact in these restricted coordinates, the local asymptotic stability follows from local positive definiteness of V_a and the La Salle invariance principle (Khalil (1996)). □

Establishing asymptotic stability of $\mathcal{Z}_a$ near $\mathcal{Z}_0$ clearly implies its forward invariance near $\mathcal{Z}_0$. Therefore it makes sense to describe the dynamics of the closed loop restricted to this set, which is easily computed by replacing $\mathbf{q}_d$ with $\mathbf{q}$ and $\boldsymbol{\omega}_d$ with $\boldsymbol{\omega}$ wherever they appear. The next step is then to prove asymptotic stability of the set

$$\mathcal{Z}_f = \{\mathbf{z} \in \mathcal{Z} \mid \mathbf{q}_\Delta = \mathbf{q}_I,\ \boldsymbol{\omega}_\Delta = \mathbf{0},\ \mathbf{f}_\Delta = 0\} \subset \mathcal{Z}_a, \tag{6.39}$$

namely the set where the virtual input $\mathbf{f}_r$ in (6.11) is the actual input of the translational dynamics (6.5). Its asymptotic stability near $\mathcal{Z}_0$, relative to initial conditions in $\mathcal{Z}_a$, is established in the following.

Lemma 6.2.3. *Set $\mathcal{Z}_f$ is globally asymptotically stable near $\mathcal{Z}_0$ for the closed-loop dynamics, relative to initial conditions in $\mathcal{Z}_a$.*

Proof. Consider the derivative of variable $\mathbf{f}_\Delta$, along dynamics (6.28)-(6.29) restricted to $\mathcal{Z}_a$ (namely such that $\mathbf{q} = \mathbf{q}_d$). Using the definition in (6.14), it holds that

$$\dot{\mathbf{f}}_\Delta = \mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\xi\dot{f} + \dot{\mathbf{R}}(\mathbf{q}_d)\mathbf{d}_*\xi f - \dot{\mathbf{f}}_r = \dot{\mathbf{f}}_{\Delta,1} + \dot{\mathbf{f}}_{\Delta,2} + \dot{\mathbf{f}}_{\Delta,3}, \tag{6.40}$$

with

$$\dot{\mathbf{f}}_{\Delta,1} = \mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\xi\dot{f} = \left(\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\right)\left(\mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\right)^\top \boldsymbol{\nu} = \mathbf{R}(\mathbf{q}_d)\mathbf{d}_*\mathbf{d}_*^\top\mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu} \tag{6.41}$$

$$\dot{\mathbf{f}}_{\Delta,2} = \dot{\mathbf{R}}(\mathbf{q}_d)\mathbf{d}_*\xi f = \mathbf{R}(\mathbf{q}_d)[\boldsymbol{\omega}_d]_\times\mathbf{d}_*\xi f = -\mathbf{R}(\mathbf{q}_d)[\mathbf{d}_*]_\times[\mathbf{d}_*]_\times\mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu} \tag{6.42}$$

$$\dot{\mathbf{f}}_{\Delta,3} = -\dot{\mathbf{f}}_r = k_{pp}\dot{\mathbf{e}}_p + k_{pd}\dot{\mathbf{e}}_v = k_{pp}\mathbf{e}_v + \frac{k_{pd}}{m}\left(-k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v + \mathbf{f}_\Delta\right) \tag{6.43}$$

where it is used the selection of $\boldsymbol{\omega}_d, \dot{f}$ in (6.15),(6.16), respectively, and $\mathbf{f}_r$ in (6.54). Hence, employing (6.17), it follows that

$$\dot{\mathbf{f}}_\Delta = \boldsymbol{\nu} - \frac{k_{pd}k_{pp}}{m}\mathbf{e}_p - \left(\frac{k_{pd}^2}{m} - k_{pp}\right)\mathbf{e}_v + \frac{k_{pd}}{m}\mathbf{f}_\Delta = -k_\Delta\mathbf{f}_\Delta. \tag{6.44}$$

Relation $\dot{\mathbf{f}}_\Delta = -k_\Delta\mathbf{f}_\Delta$ in (6.44) clearly establishes the exponential stability of $\mathcal{Z}_f$ near $\mathcal{Z}_0$ for the dynamics restricted to $\mathcal{Z}_a$, using the Lyapunov function $V_\Delta = \mathbf{f}_\Delta^2$. □

As a final step, let consider the set $\mathcal{Z}_0$ introduced in (6.31) and restrict the attention to initial conditions in the set $\mathcal{Z}_f$. The following result can be established.

Lemma 6.2.4. *Set $\mathcal{Z}_0$ is locally asymptotically stable for the closed-loop dynamics, relative to initial conditions in $\mathcal{Z}_f$.*

Proof. Let consider dynamics (6.28)-(6.29) for initial conditions in $\mathcal{Z}_f \subset \mathcal{Z}_a$. Such dynamics corresponds to the situation of input $\mathbf{f}_r$ acting directly on the translational component of the plant (6.5). As a consequence, the exponential stability is easily established by using the Lyapunov function

$$V_p = \frac{1}{2}m\mathbf{e}_v^2 + \frac{1}{2}k_{pp}\mathbf{e}_p^2, \tag{6.45}$$

for which it is easy to verify that along the dynamics restricted to $\mathcal{Z}_f$ it occurs

$$\begin{aligned}\dot{V}_p &= m\mathbf{e}_v^\top \dot{\mathbf{e}}_v + k_{pp}\mathbf{e}_p^\top \dot{\mathbf{e}}_p & (6.46)\\ &= \mathbf{e}_v^\top(-mg\mathbf{e}_3 + \mathbf{f}_r) + k_{pp}\mathbf{e}_p^\top \mathbf{e}_v & (6.47)\\ &= \mathbf{e}_v^\top(-k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v) + k_{pp}\mathbf{e}_p^\top \mathbf{e}_v & (6.48)\\ &= -k_{pd}\|\mathbf{e}_v\|^2. & (6.49)\end{aligned}$$

Applying La Salle invariance principle, the following set is locally asymptotically stable relative to $\mathcal{Z}_f$

$$\mathcal{Z}_q = \Big\{\mathbf{z} \in \mathcal{Z} \mid \mathbf{q}_\Delta = \mathbf{q}_I, \boldsymbol{\omega}_\Delta = \mathbf{0}, \mathbf{f}_\Delta = \mathbf{0}, \mathbf{e}_t = \mathbf{0}, \mathbf{q} \in \mathbb{S}^3\Big\}. \tag{6.50}$$

First note that in $\mathcal{Z}_q$, it is $\boldsymbol{\nu} = \mathbf{0}$ and $\boldsymbol{\omega}_d = \mathbf{0}$. In addition, since $\boldsymbol{\omega}_\Delta = \mathbf{0}$, it holds that $\boldsymbol{\omega} = \boldsymbol{\omega}_d = \mathbf{0}$, meaning that the attitude $\mathbf{q}$ is constant in $\mathcal{Z}_q$. Finally, using $\mathbf{f}_\Delta = \mathbf{0}$ and $\mathbf{q}_\Delta = \mathbf{q}_I$ (entailing $\mathbf{R}(\mathbf{q}) = \mathbf{R}(\mathbf{q}_d)$), it occurs that $\mathbf{R}(\mathbf{q})\mathbf{d}_*\xi f = mg\mathbf{e}_3$, which clearly implies $|f| = mg/\xi$. These derivations entail that $\mathcal{Z}_q = \mathcal{Z}_0$, thus completing the proof. □

The above stated lemmas establish a cascaded-like structure of the error dynamics composed of three hierarchically related subcomponents converging to suitable closed and forward invariant nested subsets of the space $\mathcal{Z}$ where the variable $\mathbf{z}$ in (6.30) evolves. These three closed subsets are $\mathcal{Z}_0 \subset \mathcal{Z}_f$, $\mathcal{Z}_f \subset \mathcal{Z}_a$ and $\mathcal{Z}_a \subset \mathcal{Z}$, where the smallest one $\mathcal{Z}_0$ is also compact. Such a hierarchical structure well matches the stability results established in Proposition 14 of El-Hawwary and Maggiore (2013) whose conclusion, together with the results of Lemmas 6.2.2-6.2.4 implies the following main result.

Theorem 6.2.5. *Consider the closed-loop system represented in Figure 6.1 between plant (6.5)-(6.6) and the controller presented in Section 6.2.2. The compact set $\mathcal{Z}_0$ in (6.31) is asymptotically stable for the corresponding dynamics.*

6.2.5 Extension

The control goal can be extended with an additional requirement of restricted stabilization of a given *reference orientation* $\mathbf{q}_r \in \mathbb{S}^3$ (where 'restricted' refers to the fact that such an orientation should be tracked at a lower hierarchical priority as compared to the translational error stabilization).

For this extended goal, it is possible to modify the expression of $\boldsymbol{\omega}_d$ in order to exploit all the available DoFs. Specifically, an additional term could be introduced in (6.15) to asymptotically control the platform rotation around direction $\mathbf{d}_*$, with the aim of minimizing the mismatch between $\mathbf{q}_d$ and $\mathbf{q}_r$. To this end, let consider the following quantity belonging to $\mathbb{S}^3$

$$\mathbf{q}'_\Delta = \mathbf{q}_r^{-1} \odot \mathbf{q}_d = \begin{bmatrix} \eta_r\eta_d + \boldsymbol{\epsilon}_r^\top \boldsymbol{\epsilon}_d \\ -\eta_d\boldsymbol{\epsilon}_r + \eta_r\boldsymbol{\epsilon}_d - [\boldsymbol{\epsilon}_r]_\times \boldsymbol{\epsilon}_d \end{bmatrix} = \begin{bmatrix} \eta'_\Delta \\ \boldsymbol{\epsilon}'_\Delta \end{bmatrix}. \tag{6.51}$$

Then, the extended control goal can be achieved by replacing expression (6.15) by the following alternative form

$$\boldsymbol{\omega}_d = \frac{1}{\xi f} \left[\mathbf{d}_*\right]_\times \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu} + \boldsymbol{\omega}_d' \quad \text{where} \quad \boldsymbol{\omega}_d' = k_q \mathbf{d}_* \mathbf{d}_*^\top \boldsymbol{\epsilon}_\Delta', \tag{6.52}$$

being $k_q \in \mathbb{R}^+$ a proportional gain. The pre-multiplication $\mathbf{d}_*\mathbf{d}_*^\top$ is needed to ensure that the additional term does not influence the translational dynamics, thereby encoding the hierarchical structure of the extended control goal; in other words the orientation $\mathbf{q}_r$ is obtained maintaining the translational error of the platform equal to zero. Indeed, it is easy to verify that choice (6.52) keeps expression (6.44) of $\dot{\mathbf{f}}_\Delta$ unchanged. On the other hand, it should be noted that expression (6.20) will have an additional term, once the extended version of (6.15) is considered.

The effectiveness of selection (6.52) towards restricted tracking of orientation $\mathbf{q}_r$ can be well established by using the Lyapunov function $V_\Delta' = 2(1 - \eta_\Delta')$. Indeed, following the nested proof technique based on reduction theorems, it is enough to verify the negative semi-definiteness of the Lyapunov function derivative in the set $\mathcal{Z}_0$, where $\boldsymbol{\nu} = \mathbf{0}$ and $\boldsymbol{\omega}_d = \boldsymbol{\omega}_d'$. Then, using (6.12), (6.51), (6.52), it follows that

$$\dot{V}_\Delta' = -2\dot{\eta}_\Delta' = -(\boldsymbol{\epsilon}_\Delta')^\top \boldsymbol{\omega}_d = -k_q (\boldsymbol{\epsilon}_\Delta')^\top \mathbf{d}_* \mathbf{d}_*^\top \boldsymbol{\epsilon}_\Delta' = -k_q \|\mathbf{d}_*^\top \boldsymbol{\epsilon}_\Delta'\|^2. \tag{6.53}$$

Recalling that in set $\mathcal{Z}_0$ it holds that $\mathbf{R}(\mathbf{q}_d)\mathbf{d}_* = \mathbf{e}_3$, the above analysis reveals that asymptotically one obtains $(\boldsymbol{\epsilon}_\Delta')^\top \mathbf{e}_3 = 0$, which seems to suggest that there is some control achievement (within the restricted goal) in the yaw direction.

6.2.6 Simulation Results

The effectiveness of the proposed static hovering controller is here validated by numerical simulations carried out on a particular instantiation of star-shaped hexarotor GMT.

This is a fully actuated platform which does not strictly belongs to the class described in Section 5.4.1 since all the propellers are tilted according to (5.35) setting $\alpha = \frac{7\pi}{36}, \beta = \frac{2\pi}{36}$ and $\gamma = 0$, but $\alpha_i = (-1)^{i-1}\left(1 + \frac{1}{4}\lfloor\frac{i-1}{2}\rfloor\right)\alpha$ with $i = 1 \dots 6$. Roughly speaking, along their $\overrightarrow{O_B O_{P_i}}$-axis, the consecutive propellers are opposed tilted of the same angle, however the tilt angles of the rotor pairs $(1, 2)$, $(3, 4)$ and $(5, 6)$ are different.

The choice of this complex and rather anomalous configuration is motivated by the fact that it can realize the static hovering condition and satisfies the additional condition (6.2) but the matrix $\mathbf{K}$ is not trivially the identity matrix. Nevertheless, it can be chosen as the product between an orthogonal basis of the null space of the force input matrix $\mathbf{F}$ and its transpose, which is not in the null space of the control input matrix $\mathbf{M}$.

The performed simulation exploits the dynamic model (6.5)-(6.6) extended by several real-world effects to increase the fidelity.

- Position and orientation feedback and their derivatives are impinged on time delay t_f =12 ms and sensor Gaussian noise according to Table 6.1. The actual position

$\mathbf{p}$	$\dot{\mathbf{p}}$	$\mathbf{q}/\mathbf{R}$	$\boldsymbol{\omega}$	$\mathbf{u}$
6.4e-04	1.4e-03	1.2e-03	2.7e-03	0-0.32

Table 6.1. Standard deviation of modeled sensor noise loaded on reported quantities.

and orientation are fed back with a lower sampling frequency of only 100 Hz while the controller runs at 500 Hz. These properties are reflecting a typical motion capture system and an inertial measurement unit (IMU).

- The electronic speed controller (ESC) driving the motors is simply modeled by quantizing the desired input $\mathbf{u}$ resembling a 10 bit discretization in the feasible motor speed resulting in a step size of ≈ 0.12 Hz. Additionally the motor-propeller combination is modeled as a first order transfer function $\left(G(s) = \frac{1}{1+0.005s}\right)$. The resulting signal is loaded again with a rotational velocity dependent Gaussian noise (see Table 6.1). This combination mirrors with high realism the dynamic behavior of a common ESC motor-propeller combination, i.e., BL-Ctrl-2.0, by MikroKopter, Robbe ROXXY 2827-35 and a 10 inch rotor blade (Franchi and Mallet (2017)).

- In normal working conditions, the spinning rate of all the propellers is assumed to be constrained within the feasible frequencies set $\mathcal{U} = [0, 120]$ Hz. These limitations are realistic for 12 inch rotors actuated by MK3638 brushless motors by MikroKopter (propeller-motor combination used in the real platform described in Section 6.3.3).

The control goal is to steer the described vehicle to a locally stable equilibrium position $\mathbf{p}_r \in \mathbb{R}^3$ starting from the initial position which is fixed at the origin of the inertial world frame. In addition, the stabilization of the platform attitude is required without imposing a reference orientation.

The simulation results are depicted in Figure 6.2. The first and second plot report the position and the orientation of the hexarotor, respectively. The roll-pitch-yaw angles (ϕ, θ, ψ) are used to represent the attitude in order to give a better insight of the vehicle behavior; however, the internal computations are all done with unit quaternions. Note that the hexarotor smoothly achieves the reference position in roughly 5s. After this transient, the position error $\mathbf{e}_p$ (third plot) converges to zero. This behavior is expected in light of the robustness results of asymptotic stability of compact attractors, established in Goebel, Sanfelice, and Teel (2012). On the other hand, the orientation $\mathbf{q}$ of the vehicle converges to the desired one $\mathbf{q}_d$ with a comparable transient period. This is clearly observable also in the fourth plot that reports the trend of the roll-pitch-yaw angles associated to $\mathbf{q}_\Delta$. Note that $\mathbf{q}_d = \mathbf{q}_I$. For the sake of completeness, the last plot in Figure 6.2 illustrates the control inputs provided to each propeller: at the steady-state all the spinning rates are included in $[80, 110]$Hz that represents a feasible range of values from the practical point of view. In this sense, it is possible to conclude that using this controller the static hovering feasibility introduced in Definition 5.3.10 is guaranteed for the considered hexarotor GTM.

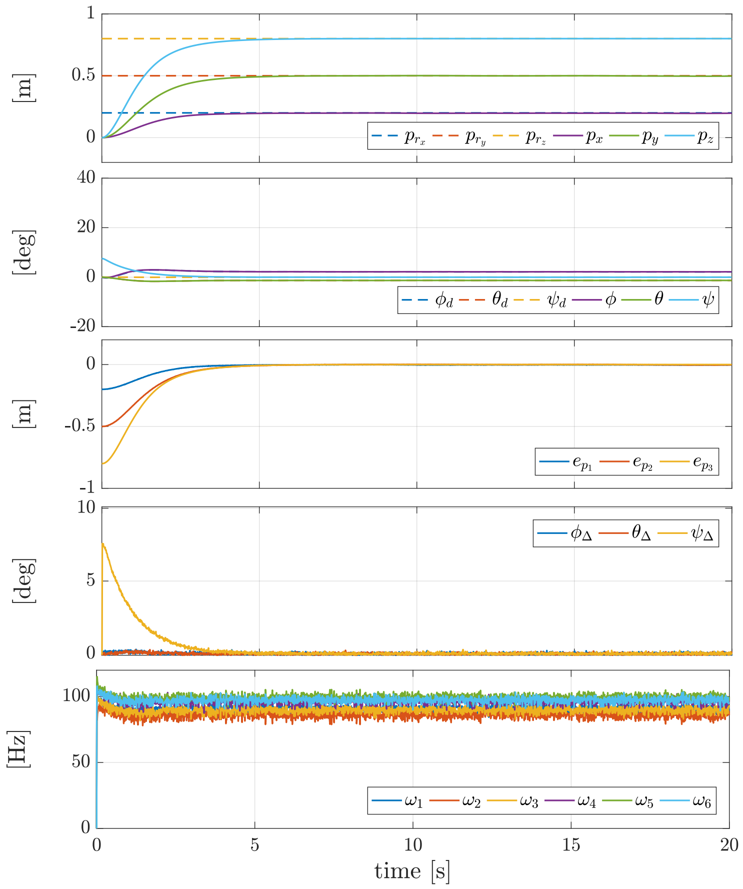


Figure 6.2. Realistic simulation of the control of a particular instantiation of tilted star-shaped hexarotor GTM by using the proposed feedback controller.

Figure 6.3 shows the performance of the static hovering controller when also a constant given orientation is required. The error trends and the commanded spinning rates are comparable with the previous case, while the second plot demonstrates that the hexarotor rotates until it achieves the given $\mathbf{q}_r$. A very small bias ($\approx 2°$) is observable in the roll and pitch components. Nevertheless, the fourth plot ensures that these at least converge towards the desired values: the roll-pitch-yaw angles related to $\mathbf{q}_\Delta$ converge towards zero guaranteeing that the current orientation $\mathbf{q}$ approximates the desired one $\mathbf{q}_d$, which results to be slightly different from the required $\mathbf{q}_r$.

6.3 Cascaded Geometric Controller

In this section a second solution is described to steer and keep a statically hoverable GTM platform (namely, satisfying conditions (5.11),(5.28),(5.30)) at a given constant reference position $\mathbf{p}_r \in \mathbb{R}^3$ while keeping fixed its attitude. The dynamics of the vehicle is governed by (5.9)-(5.10): its orientation is again represented using the rotation matrix formalism. In addition, here the hovering orientation is also required to be as close as possible to a given reference one defined by the constant rotation matrix $\mathbf{R}_r \in SO(3)$.

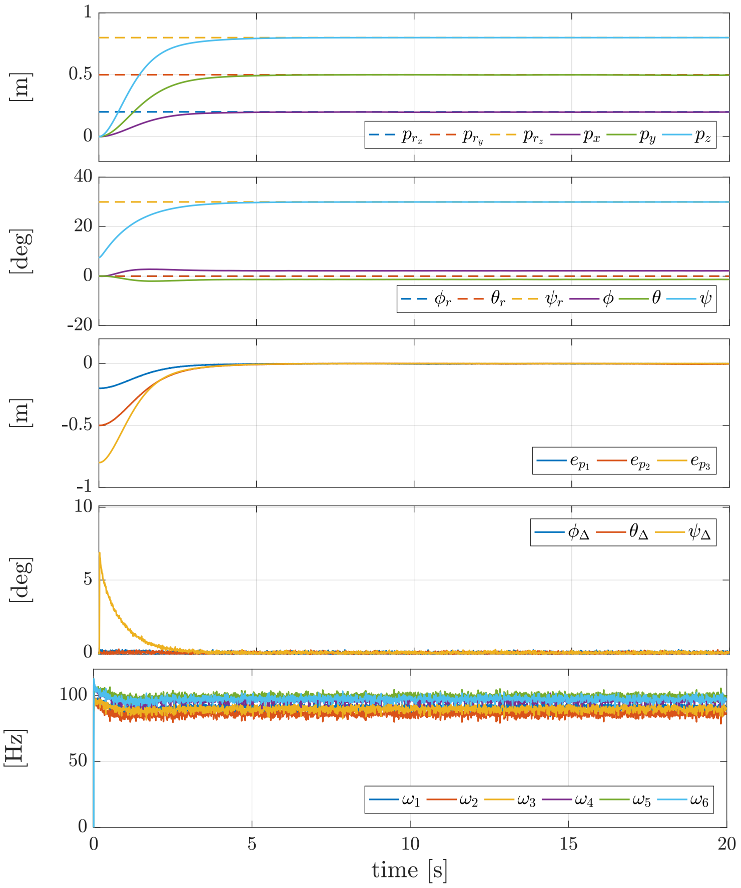


Figure 6.3. Realistic simulation of the control of a particular instantiation of tilted star-shaped hexarotor GTM by using the proposed feedback controller and providing a constant reference orientation.

Employing a cascaded scheme (Figure 6.4) this controller is composed by an outer control loop (*position controller*) computing the reference control force $\mathbf{f}_r$ and the desired orientation $\mathbf{R}_d$, which is passed to an inner control loop (*attitude controller*) outputting the reference control moment $\boldsymbol{\tau}_r$. The actual input is computed by a wrench mapper that determines $\mathbf{u}$ so that $\boldsymbol{\tau}_c = \mathbf{M}\mathbf{u} = \boldsymbol{\tau}_r$ (i.e., guaranteeing the achievement of the reference control moment as in the previous control solution) and, at least, the projection of the control force $\mathbf{f}_c = \mathbf{F}\mathbf{u}$ along the preferential direction $\mathbf{d}_*$ is equal to the norm of $\mathbf{f}_r$. The structure of the proposed controller thus consists of the three main blocks described in the following. Note that, as mentioned in Section 6.1.2, the preferential direction $\mathbf{d}_*$ constitutes a key feature also of this control strategy and it is selected as in (5.33).

Position Controller

Given the constant reference position $\mathbf{p}_r$, a good choice to steer the position error $\mathbf{e}_p = \mathbf{p} - \mathbf{p}_r$ (and the velocity error $\mathbf{e}_v = \dot{\mathbf{p}}$) to zero consists in applying to the vehicle the (reference) force

$$\mathbf{f}_r = mg\mathbf{e}_3 - \mathbf{K}_p\mathbf{e}_p - \mathbf{K}_v\mathbf{e}_v, \tag{6.54}$$

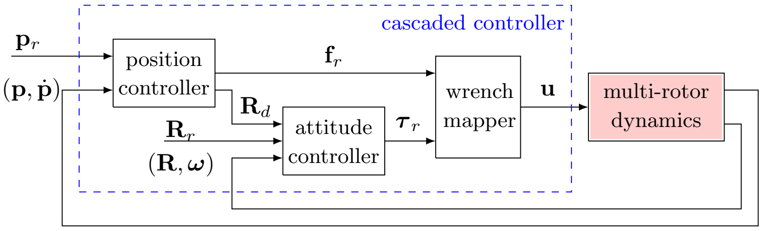


Figure 6.4. Block diagram of the closed-loop system with the proposed cascaded control strategy.

which is a simple PD + gravity compensation control as in (6.11) with positive definite diagonal gain matrices $\mathbf{K}_p, \mathbf{K}_v \in \mathbb{R}^{3\times3}$ in place of scalar gains. This substitution allows to (potentially) increase both the control modulation over the different components and the effort in terms of number of parameters (controller gains) to tune. For the non-linear controller in Section 6.2, the introduction of scalar quantities is instead mainly motivated by the notation compactness.

As in the previous solution, the fundamental idea behind the cascaded controller is to obtain $\mathbf{f}_c = \mathbf{f}_r$ by rotating the platform in such a way that the preferential direction $\mathbf{d}_*$ is aligned with $\mathbf{f}_r$. In fact in this way, both the direction and the intensity of $\mathbf{f}_c$ can be made equal to the direction and intensity of $\mathbf{f}_r$ regardless of the implemented moment $\boldsymbol{\tau}_c$. To do so the position controller computes the desired orientation $\mathbf{R}_d \in SO(3)$ that carries out the alignment of the preferential direction to the direction of the required control force, i.e., $\mathbf{R}_d$ is such that $\mathbf{R}_d\mathbf{d}_* = \mathbf{f}_r/\|\mathbf{f}_r\|_2$ (as in the previous case). Nevertheless, differently from the controller in Section 6.2, the desired orientation is here determined also trying to minimize the distance from the reference orientation $\mathbf{R}_r$.

As a consequence, the matrix $\mathbf{R}_d$ is univocally determined among the infinite possible solutions by first computing the minimal rotation $\mathbf{R}_b \in SO(3)$ that aligns the preferential direction and the third inertial axis $\mathbf{e}_3$. This is possible through the Rodrigues' rotation formula (Bauchau and Trainelli (2003)) so that

$$\mathbf{R}_b = \mathbf{I}_3 + [\mathbf{b}]_\times + [\mathbf{b}]_\times \frac{1-a}{\|\mathbf{b}\|_2}, \tag{6.55}$$

where $\mathbf{b} = [\mathbf{d}_*]_\times \mathbf{e}_3 \in \mathbb{R}^3$ and $a = \mathbf{d}_*^\top \mathbf{e}_3 \in \mathbb{R}$ are respectively the cross and the inner product of the preferential direction and the third inertial axis. Then, a matrix $\mathbf{R}_w \in SO(3)$ is defined in order to fix one of the infinite rotations that align $\mathbf{e}_3$ to the normalized vector $\mathbf{f}_r/\|\mathbf{f}_r\|_2$. To do so, the requirement on minimal distance from $\mathbf{R}_r$ is exploited. Hence, using the approach in Lee et al. (2010), it is set

$$\mathbf{R}_w = [((\mathbf{r}_3 \times \mathbf{r}_1) \times \mathbf{r}_3) \;\; (\mathbf{r}_3 \times \mathbf{r}_1) \;\; \mathbf{r}_3], \tag{6.56}$$

where $\mathbf{r}_1 \in \mathbb{R}^3$ is the first column of $\mathbf{R}_r$ and $\mathbf{r}_3 = \mathbf{f}_r/\|\mathbf{f}_r\|_3 \in \mathbb{R}^3$. The desired orientation is univocally determined by the product

$$\mathbf{R}_d = \mathbf{R}_w \mathbf{R}_b. \tag{6.57}$$

Attitude Controller

To let the current attitude $\mathbf{R}$ track the desired orientation $\mathbf{R}_d$, the standard geometric control moment is computed as

$$\boldsymbol{\tau}_r = \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} - \mathbf{K}_R \mathbf{e}_R - \mathbf{K}_\omega \boldsymbol{\omega}, \tag{6.58}$$

where the positive definite diagonal gain matrices $\mathbf{K}_R, \mathbf{K}_\omega \in \mathbb{R}^{3\times 3}$ allow to tune the effect of the rotation tracking error $\mathbf{e}_R$ and the angular velocity feedback term, respectively. Specifically, the rotation tracking error is chosen to be

$$\mathbf{e}_R = \frac{1}{2}(\mathbf{R}_d^\top \mathbf{R} - \mathbf{R}^\top \mathbf{R}_d)^\vee, \tag{6.59}$$

where the operator $[\cdot]^\vee$ describes the map from $\mathfrak{so}(3)$ to $\mathbb{R}^3$ that associates a skew-symmetric matrix to the corresponding three-dimensional non-zero vector.

Remark 6.3.1. Beside the similarities, the expression (6.58) differs from (6.19) because of the lack of the feedforward term. From this perspective, the cascaded controller represents a more viable solution due to the fact that $\boldsymbol{\omega}_d$ and $\dot{\boldsymbol{\omega}}_d$ in (6.19) are not measurable quantities and have to be numerically retrieved under real-time constraints.

Wrench Mapper

The proposed control law prioritizes the fulfillment of the required control torque, namely to obtain $\boldsymbol{\tau}_c = \boldsymbol{\tau}_r$. To do so the actual control input is designed as the sum of two terms:

$$\mathbf{u} = \mathbf{M}^\dagger \boldsymbol{\tau}_r + \bar{\mathbf{u}} f. \tag{6.60}$$

Note that, with respect to (6.8), here the control input vector is computed using the pseudo-inverse $\mathbf{M}^\dagger = \mathbf{M}^\top(\mathbf{M}\mathbf{M}^\top)^{-1} \in \mathbb{R}^{n\times n}$ in place of its generalized version. This is justified by the fact that the additional condition (6.2) is not considered here; in other words, any assumption is in place about the decoupling of the control.

The value of the parameter $f \in \mathbb{R}$ is set requiring that the total force applied to the body, i.e., $\mathbf{RFu}$, is made as close as possible to $\mathbf{f}_r$, thus solving

$$\min_f \|\mathbf{RF}(\mathbf{M}^\dagger \boldsymbol{\tau}_r + \bar{\mathbf{u}} f) - \mathbf{f}_r\|^2, \tag{6.61}$$

whose solution in closed form is obtained through the simple vector projection using the formula

$$f = \frac{(\mathbf{f}_r - \mathbf{RFM}^\dagger \boldsymbol{\tau}_r)^T \mathbf{RF}\bar{\mathbf{u}}}{\|\mathbf{RFb}_2\|^2} = \frac{1}{\xi}(\mathbf{f}_r - \mathbf{RFM}^\dagger \boldsymbol{\tau}_r)^T \mathbf{Rd}_*. \tag{6.62}$$

As a consequence, the total input control law $\mathbf{u}$ results to be

$$\mathbf{u} = \mathbf{M}^\dagger \boldsymbol{\tau}_r + \frac{1}{\xi}\bar{\mathbf{u}}(\mathbf{f}_r - \mathbf{RFM}^\dagger \boldsymbol{\tau}_r)^T \mathbf{Rd}_*. \tag{6.63}$$

Note that as long as $\mathbf{Rd}_*$ is kept parallel to $\mathbf{f}_r$ then $\boldsymbol{\tau}_r = \mathbf{0}$, and therefore it follows that the total force is $\frac{1}{\xi}\mathbf{RF}\bar{\mathbf{u}}\mathbf{f}_r^T\mathbf{Rd}_* = \mathbf{Rd}_*\mathbf{f}_r^T\mathbf{Rd}_* = \mathbf{f}_r$ as it should be.

Remark 6.3.2. This control law ensures $\boldsymbol{\tau}_c = \boldsymbol{\tau}_r$. Moreover, it occurs that $\mathbf{f}_c = \mathbf{d}_*\|\mathbf{f}_r\| + (\mathbf{I}_3 - \mathbf{d}_*\mathbf{d}_*^\top)\mathbf{FM}^\dagger\boldsymbol{\tau}_r = \mathbf{d}_*\|\mathbf{f}_r\| + P(\mathbf{d}_*)\mathbf{FM}^\dagger\boldsymbol{\tau}_r$ where $P : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ is the *orthogonal projector operator* which geometrically projects any non-zero vector onto its orthogonal complement. Therefore the more the attitude control (6.59) is able to make $\mathbf{d}_*$ aligned with $\mathbf{f}_r$ (and, as a consequence, to let $\boldsymbol{\tau}_r$ converge to $\mathbf{0}$) the more $\mathbf{f}_c$ converges to $\mathbf{f}_r$, which is what one aims to. However, during the transient, the component of the control force along the direction perpendicular to $\mathbf{d}_*$ cannot be assigned at will, and actually depends by $\boldsymbol{\tau}_r$. This is due to the fact that the decoupling property described in Section 5.2 is missing. In this sense, the presented controller represents a generalization of the cascaded control to non-decoupled cases. Indeed, when the decoupling property holds the component of the control force along the direction perpendicular to $\mathbf{d}_*$ will be forced to be zero and the platform will behave like a standard under-actuated multi-rotor subject to a standard cascaded control.

6.3.1 Stability Analysis

The asymptotic convergence of the errors dynamics is not analytically provable in this context mainly because of the expression of $\boldsymbol{\tau}_r$ in (6.19) which implements any feedforward term aiming at ruling out the non-linear part of the platform orientation dynamics.

The strong non-linearity of the motion equations (5.9)-(5.10) is thus the principal obstacle towards a formal convergence proof. Nevertheless, the effectiveness of the proposed controller emerges from the results of the extensive campaign of simulative and experimental tests. These have been performed taking into account the critic situation of propellers loss for different kinds of hexarotor GTMs. The choice of considering a failed scenario is motivated by two reasons. On one side, after the loss of a rotor, the considered vehicles result to be under-actuated platforms whose preferential direction is not perpendicular to the ground: this framework strongly deviates from the standard one (as anticipated at the beginning of the chapter). On the other hand, the performed tests validate also the results obtained on the robustness analysis conducted in Section 5.4.

6.3.2 Simulation Results

This section illustrates the simulation results for two different platforms utilizing the cascaded geometric controller. First, a $\left(\frac{7\pi}{36}, \frac{5\pi}{36}, 0\right)$-hexarotor GTM is taken into account, namely an instantiation of a star-shaped hexarotor whose propellers are tilted with $\alpha = 35°$ and $\beta = 25°$[1]. Then a Y-shaped hexarotor (namely a $(0, 0, \frac{\pi}{3})$-hexarotor GTM) is considered. In order to carry out a realistic simulation, for both vehicles the same non-idealities introduced in Section 6.2.6 about the dynamic models are assumed.

[1] These angles represent a good choice to achieve a balance between full actuation and inefficient losses as a result of internal forces.

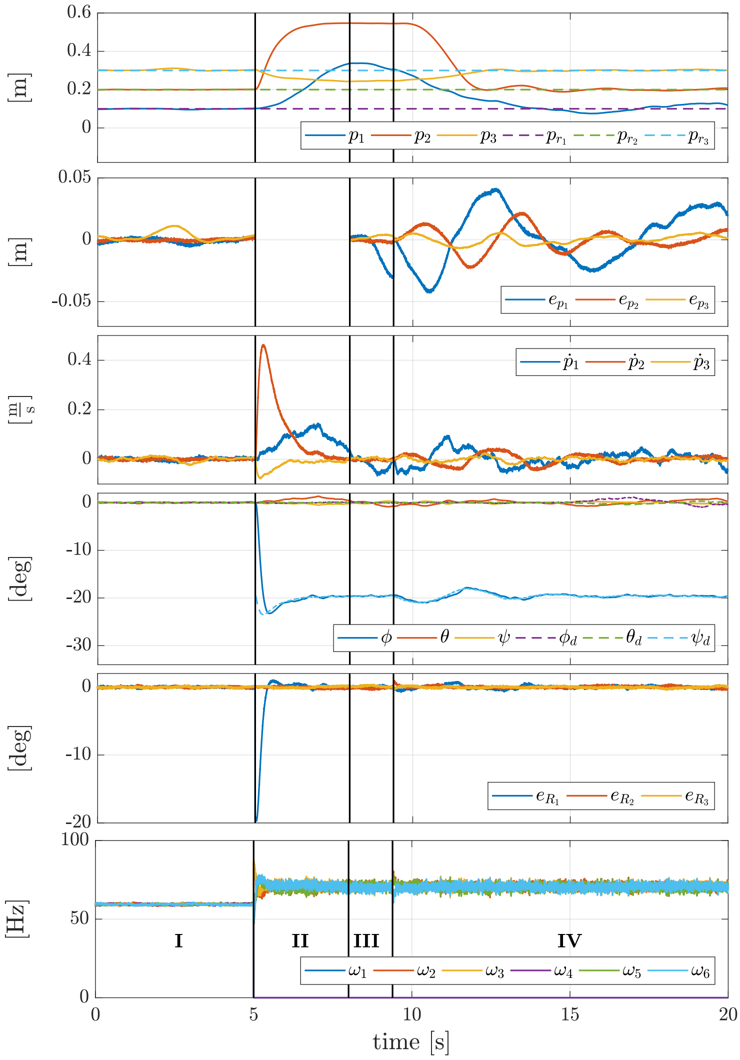


Figure 6.5. Realistic simulation of the control of a $\left(\frac{7\pi}{36}, \frac{5\pi}{36}, 0\right)$–hexarotor GTM in the case of a single motor failure by using the proposed cascaded controller.

The controller is tested in case of a rotor-failure: in both the simulated scenarios the vehicle shall hover at a predefined spot $\mathbf{p}_r \in \mathbb{R}^3$ and $\mathbf{R}_r \in SO(3)$ fulfilling a surveillance task; at time $t = t^*$ the failure of a single rotor is assumed and the cascaded geometric controller is utilized to recover from this threatening situation. In the first test, the failure is assumed to be instantaneous, i.e., the spinning rate of the failed propeller immediately goes to zero at $t = t^*$, while in the second test the failed propeller is assumed to slowly degrades its performance, generating a force that decreases until zero with an exponential decay having half-life time[2] $t_{\frac{1}{2}} = 0.1\,\mathrm{s}$. This last assumption adds an unknown force and torque disturbance in the moment of failure.

Tilted Star-shaped Hexarotor

The results of the simulation for the $\left(\frac{7\pi}{36}, \frac{5\pi}{36}, 0\right)$-hexarotor GTM are reported in Figure 6.5 where the vertical lines identify different phases of the test.

[2] The *half-life time* is the time required for the decaying quantity to fall to one half of its initial value.

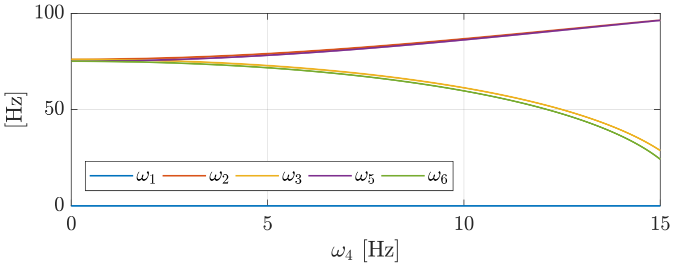


Figure 6.6. Spinning rates at hovering equilibrium when ω_4 changes in $[0, 15]$ Hz.

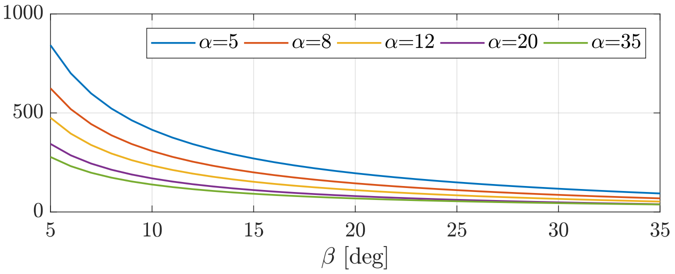


Figure 6.7. Condition number of matrix obtained from $\mathbf{H}\left(\frac{7\pi}{36}, \frac{5\pi}{36}, 0\right)$ once by removing the first and fourth columns with respect to β angle for different values (in degrees) of α angle.

At the beginning (*phase I*), the vehicle hovers in a given constant reference position $\mathbf{p}_r$, while its attitude is kept equal to $\mathbf{R}_r$. In this initial safe state, both the position and orientation errors are negligible (second and fifth plots). At $t^* = 5\,\mathrm{s}$, the instantaneous failure of the rotor 1 is simulated: its spinning rate falls to zero. Simultaneously, the opposing propeller is switched off. This action is not strictly mandatory, however selecting $\omega_4 = 0\,\mathrm{Hz}$ represents the best solution for the purpose of balancing the required control effort. This is clear observing Figure 6.6 that depicts the spinning rates commanded at the hovering equilibrium when the propeller 1 is failed (i.e., $\omega_1 = 0\,\mathrm{Hz}$) for different values of ω_4. Switching off the 4-th motor all the remaining healthy propellers are required to spin at the same velocity, thus balancing the control effort.

Note that at the failure instant t^* =5 s, the controller asks for a new hovering orientation as visible by the fourth plot in Figure 6.5 that reports the current and desired orientation expressed in terms of roll-pitch-yaw angles. This discontinuity in the steady hovering orientation is due to the presence of non-zero tilt angle α. Setting $\alpha \neq 0$ implies that, when any motor fails, the partially coupled resulting platform has a preferential direction which is no more perpendicular to the ground. However, this also implies a smaller condition number for the matrix that has to be inverted in the computation of the input required to achieve the control moment (6.58). In detail, denoting by $\mathbf{H}\left(\frac{7\pi}{36}, \frac{5\pi}{36}, 0\right)$ the control moment input matrix of the (healthy) platform (according to the notation in Section 5.4), to determine $\mathbf{u}$ so that $\boldsymbol{\tau}_r = \boldsymbol{\tau}_c$ it is necessary to compute the pseudo-inverse of the matrix that maps the command input vector to the control moment. This coincides with the matrix obtained from $\mathbf{H}\left(\frac{7\pi}{36}, \frac{5\pi}{36}, 0\right)$ once by removing the k-th and j-th columns, where k is the ID of the failed rotor and $j = (k + 2) \bmod 6 + 1$ refers to the opposed one. This last observation is justified by Figure 6.7, where it has been assumed $k = 1$.

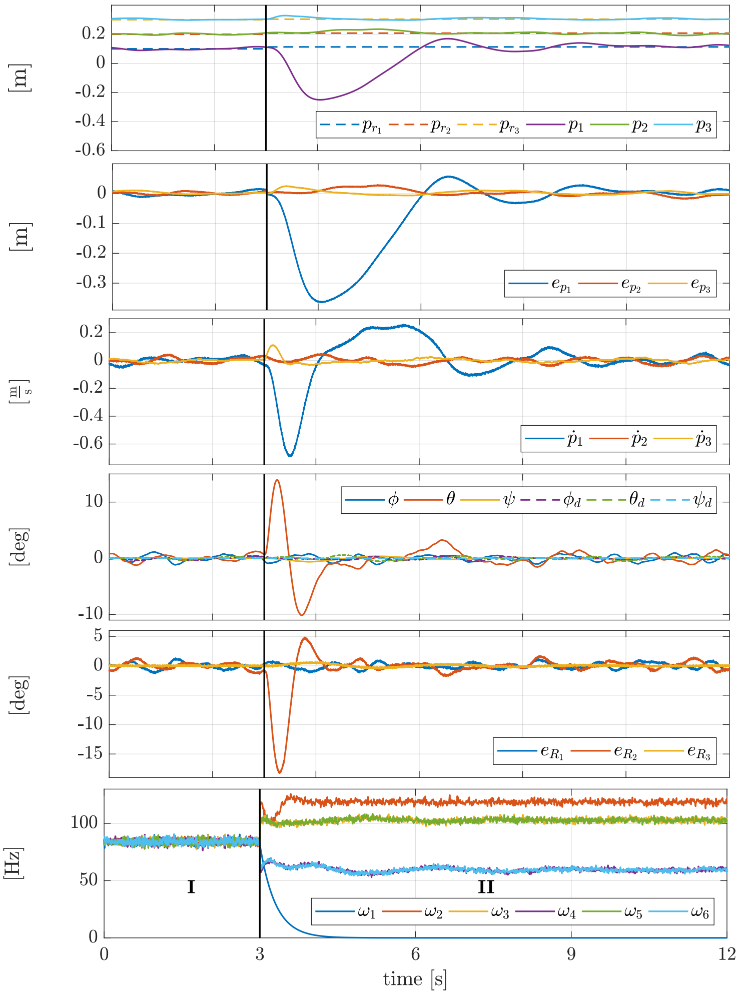


Figure 6.8. Realistic simulation of the control of a $\left(0,0,\frac{\pi}{3}\right)$–hexarotor GTM in the case of a single motor failure by using the proposed cascaded controller.

Once the failure is occurred ($t >$ 5s), for a certain time (*phase II*), the gain matrix $\mathbf{K}_p$ in (6.54) is set to $\mathbf{0} \in \mathbb{R}^{3\times3}$ allowing that the (new) orientation stabilizes and the velocity decreases (third and fourth plot). After ≈3 s, $\mathbf{K}_p$ is increased using the current position in that time instant as future desired position (*phase III*). Finally (*phase IV*), when the norm of the current velocity decreases under a certain threshold a new trajectory is determined from the current position to the original hovering position and the the platform successfully returns back.

Note that at the steady-state the commanded spinning rates of the healthy (spinning) propellers span in the range $[60, 80]$Hz. The required control effort is so that the static hovering results to be a feasible flight condition for the failed platform under the assumption in Section 6.2.6. In particular, feasibility is ensured according to the Definition 5.3.11 of *relaxed* static hovering feasibility by choosing $\mathcal{C} = \{4\}$ and $u_4{*} = 0\,$Hz.

Y-shaped Hexarotor

The results of Y-shaped hexarotor simulation are reported in Figure 6.8: in this case only two phases can be distinguished, i.e., the safe one (*phase I*) and the failed one (*phase II*).

Before the failure occurs, the position and orientation error (second and fourth plots) are negligible: the hexarotor hovers perfectly at its desired spot. Accordingly, the translational velocity (third plot) is very small (considering the realistic factors introduced in the simulation) with $\|\dot{\mathbf{p}}\|_2 < 0.07\,\mathrm{m\,s^{-1}}$. At $t^* = 3\,\mathrm{s}$, the failure of propeller 1 is simulated: ω_1 starts to exponentially decrease and the system is clearly perturbed. Immediately the translational velocity and the position errors start increasing, reaching a peak position error of $\|\mathbf{e}_p\|_2 = 0.46\,\mathrm{m}$ 1.1 s after the failure. Subsequent, the position and orientation errors decrease fast and the hexarotor GTM tracks again well the reference position. Note how propellers $2, 3$ and 5 are compensating the loss of generated thrust while propellers 4 and 6 are commanded to decrease their thrust (last plot of Figure 6.8). In any case the commanded spinning rates ensure the fulfillment of the static hovering feasibility.

6.3.3 Experimental Tests

This section presents and discusses the real-world experiments that have been conducted on a star-shaped hexarotor platform available at LAAS-CNRS[3], the Tilt-Hex.

Experimental Setup

Tilt-Hex is a fully actuated (and fully decoupled) aerial vehicle, developed at LAAS-CNRS. It is a $\left(\frac{7\pi}{36}, \frac{5\pi}{36}, 0\right)$-hexarotor GTM, namely an instantiation of a star-shaped hexarotor whose propellers are tilted with $\alpha = 35°$ and $\beta = 25°$ (same platform described in Section 6.3.2). All the mechanical parts of Tilt-Hex are off-the-shelf available or 3D printable. The diameter of the platform, including the propeller blades, is 1.05 m and the total mass, with a 2200 mA h Li-Po battery, results as $m = 1.8\,\mathrm{kg}$. MK3638 brushless motors by MikroKopter are used, together with 12 inch propeller blades to actuate Tilt-Hex. A single propeller-motor combination can provide a maximum thrust of 12 N. The ESC, a Bl-Ctrl-2.0, is as well purchased from MikroKopter. The control software running on the ESC, developed by LAAS, controls the rotational propeller speed in closed loop and additionally allows to read the current spinning rate (Franchi and Mallet (2017)). An on-board IMU provides measurements of 3 gyroscopes and a 3D accelerometer at 500 Hz. An external motion capture system (OptiTrack) provides position and orientation data at 100 Hz. These information are fused via an Unscented Kalman Filter (UKF) state estimator to obtain the full vehicle state at 500 Hz.

The controller is implemented in Matlab-Simulink and runs at 500 Hz on a stationary workstation. As its computational effort is very low (considerably below 1 ms per control loop) it could be ported easily to an on-board system. Based on the acquired experience with a similar porting, the performances of the on-board implementation would be better than the current implementation, thanks to the possibility of reaching a faster control frequency (greater than 1 kHz) and almost real-time capabilities (latency below 1 ms). Hence the performed experiments represent a worst case scenario from this point of view.

[3] LAAS-CNRS, Laboratoire d'analyse et d'architecture des systèmes, Toulouse, France.

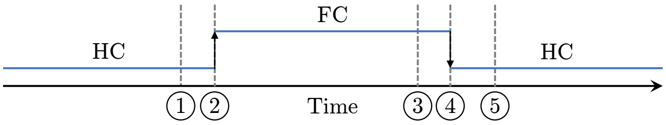


Figure 6.9. Time line of controller switching: (1) HC is running, failure is manually triggered (i-th propeller stops), (2) failure is detected, opposed propeller is stopped, controller is switched to FC, (3) stopped motors are restarted via manual trigger, (4) these rotors reach 16 Hz, controller is switched back to HC, (5) reference trajectory reaches the initial pose of Tilt-Hex.

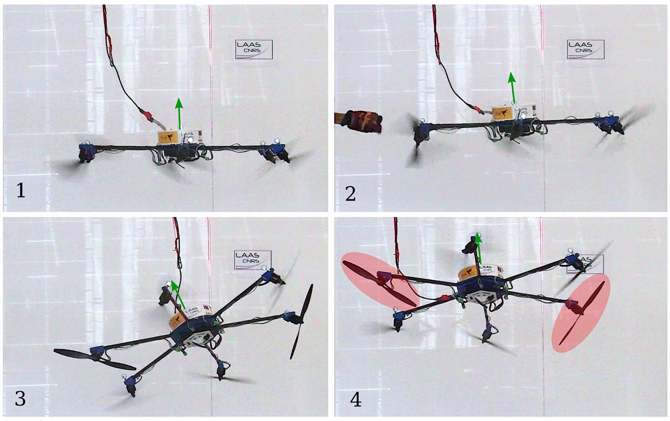


Figure 6.10. Tilt-Hex recovering from manual propeller stop: (1) static hover in healthy conditions, (2) manual stop of a propeller, (3) transient phase, (4) static hovering in failed conditions (stopped rotors are highlighted).

During the execution of all experiments two controllers have been utilized (see Figure 6.9). While Tilt-Hex is healthy (all rotors working) or before a failure detection of an ESC, the controller presented in Ryll et al. (2016) is used - referred as *Healthy Controller* (HC). As soon as a failure is detected the controller is switched to the cascaded controller described in Section 6.3 - referred as *Failed Controller* (FC). In some of the following experiments it has been manually asked a single ESC to fail. This fail request triggers the ESC to immediately stop one propeller from spinning and to rise a failure flag. The status of the failure flags of all ESCs is checked every 10 ms. When a failure is detected, the opposed propeller is stopped and the controller is switched to FC. To change back the status from failed to healthy, the two stopped motors need to be restarted. As the time duration is not always identical, the FC is used until a spinning velocity of 16 Hz (minimum closed loop spinning velocity of the ESC) is reached on both the previously stopped motors. Then the controller is switched to HC and a trajectory is computed to drive back the platform from its current position and orientation to the initial reference position and orientation smoothly. Finally Tilt-Hex reaches its initial pose.

Although in the next tests the propellers have been ‘manually’ failed for conducting several experiments in a row and in a repeatable way, the controller has been also tested by mechanically stopping a propeller through the impact with an external object during flight. Fig. 6.10 reports some significant frames of this experiment that shows the robustness of the proposed approach and the possibility of using it within a pipeline of failure detection, isolation, and reaction.

Basic principles

In the first experiment (Experiment 1) the basic principles and behavior of the controller and its recovering capabilities are presented. The results are reported of three consecutive failures of the first three propellers, resulting in the stopping of all the opposed rotor pairs (1-4, 2-5, 3-6) of Tilt-Hex. To perform the experiment the platform has been forced to recover from the failed situation restarting the failed and the actively stopped motors (according to the procedure explained in Figure 6.9). As Tilt-Hex is a fully actuated aerial vehicle a smooth transient trajectory is followed to recover the initial pose after the motor failure phase.

The results of Experiment 1 are presented in Figure 6.11. The background colors of the plots indicate the used controller: in green shaded areas HC is used, in red shaded areas FC is used while in white shaded areas FC is used as well but the two stopped motors have been already restarted. The first two plots of Figure 6.11 present the reference position $\mathbf{p}_r$ and the actual position $\mathbf{p}$, and the position error $\mathbf{e}_p$ irrespective of the used controller. Note that at the beginning, while HC is used (healthy conditions), the reference position is tracked perfectly and the platform is oriented according to the provided reference attitude.

At $t_1 = 7.58\,\mathrm{s}$ the failure of motor 1 is triggered (corresponding to event 1 in Figure 6.9) and at $t_2 = 7.6\,\mathrm{s}$ the controller is switched to FC and the opposed motor 4 is stopped (corresponding to event 2 in Figure 6.9). Immediately the position error increases, reaching a peak position error norm of $\|\mathbf{e}_p\|_2 = 0.37\,\mathrm{m}$. In the moment of controller switching a discontinuity of the reference orientation $\mathbf{R}_r$ occurs. This is evident comparing the third and fourth plot of Figure 6.11: the third plot reports the current and reference orientation expressed in terms of roll-pitch-yaw angles, while the fourth plot depicts the orientation error defined in (6.59). The discontinuity is explained by the different steady-state hovering orientations of the failed system which is due to the presence of non-zero tilt angle α, as explained in Section 6.3.2.

After the controller switching the system stabilizes within a few seconds (observe the components of $\dot{\mathbf{p}}$ and $\boldsymbol{\omega}$ in the fifth and sixth plots of Figure 6.11). The final orientation error is negligible, while a small steady-state position error (less than $0.15\,\mathrm{m}$ for each components) is still visible, which can be easily explained by the unavoidable uncertainty in the control force and moment input matrices in (5.8). This error could be further decreased using integral terms or adaptive control techniques, however the main goal here was to show that static hovering (i.e., with zero velocities) is achieved, rather than showing extremely accurate position control.

At $t_3 = 23.7\,\mathrm{s}$ the two stopped motors (1 and 4) are asked to start again and at $t_4 = 24.8\,\mathrm{s}$ both rotors are spinning with the minimum spinning velocity $\omega_{1,4} = 16\,\mathrm{Hz}$ of the ESC. The controller is switched to HC and the initial position and orientation is reached fast without any visual steady-state error.

The same procedure is repeated for motor 2 and 3. In the three failed phases different motor pairs are stopped: it is interesting to notice the different hovering orientations during the different failures.

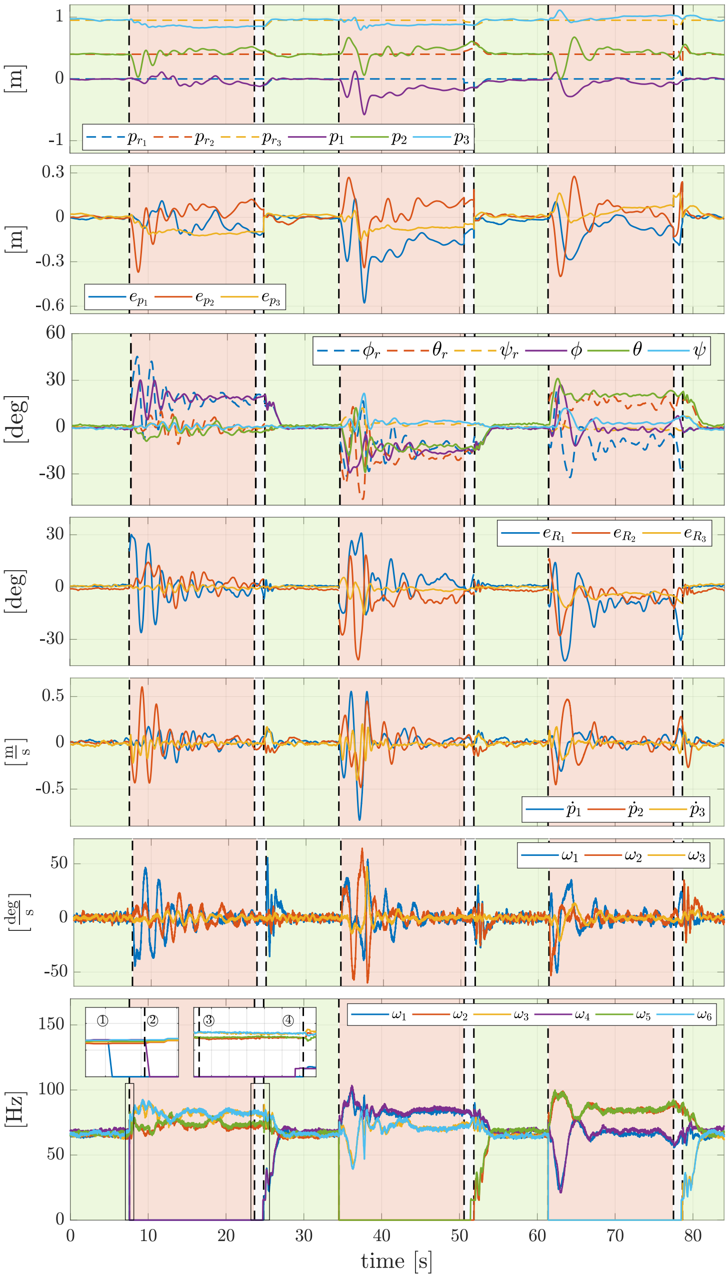


Figure 6.11. Experiment 1 - three consecutive failures and recoveries (from motor 1 till motor 3). Green shaded background - Tilt-Hex healthy; Red shaded background - Tilt-Hex rotor failed; White shaded background - failed controller but stopped propellers are already restarted.

Robustness

The robustness of the controller is tested through three experiments. Experiment 2-1 presents the accumulated results of $n = 23$ repeated failures of motor 3 and Experiment 2-2 shows the response of the system in case of a step in the reference position under failed conditions. Finally, Experiment 2-3 illustrates the response of the system to a continuously changing reference (similar to a ramp response).

In Experiment 2-1 the last phase of Experiment 1 (from 60 s to 80 s) has been repeated for 23 trials: Tilt-Hex has recovered from the failure in all the cases. To get a better

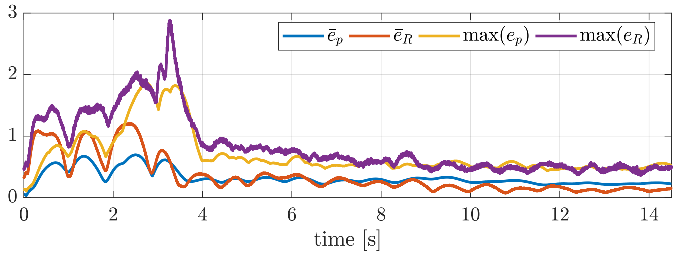


Figure 6.12. Experiment 2-1 - Mean of state error function for 23 trials in position (blue) and orientation (red) and maximum state error values over all trials (yellow and purple).

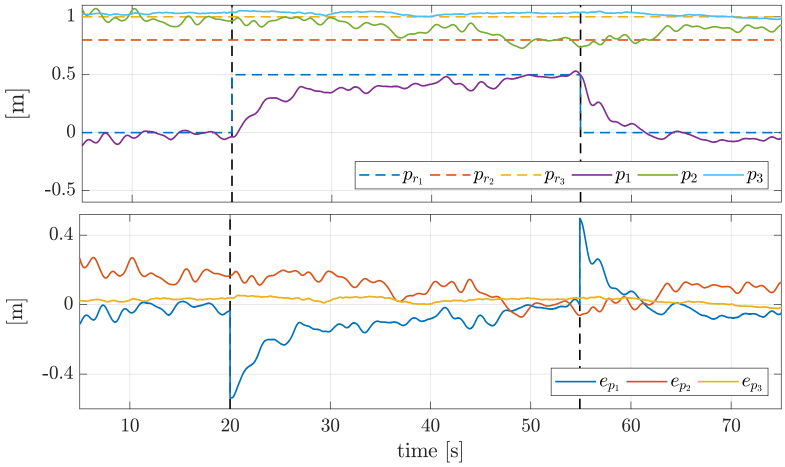


Figure 6.13. Experiment 2-2 - Step response of the system with failure. At $t = 20\,\mathrm{s}$ a step of $0.5\,\mathrm{m}$ along p_{r_1} is commanded. At $t = 55\,\mathrm{s}$ a step of $-0.5\,\mathrm{m}$ along p_{r_1} is commanded. The time of the step signal is indicated by the black dashed line in all plots.

understanding of the vehicle performance, a new position and orientation error function are defined representing the state error

$$e_p = \|\mathbf{e}_p\|_2 + k\|\mathbf{e}_v\|_2,$$
$$e_R = \|\mathbf{e}_R\|_2 + k\|\mathbf{e}_\omega\|_2,$$

with $k = 1\,\mathrm{s}$, $\mathbf{e}_v = \dot{\mathbf{p}}$ and $\mathbf{e}_\omega = \boldsymbol{\omega}$. Figure 6.12 reports the mean error value $\bar{e}_p \in \mathbb{R}_0^+$ and $\bar{e}_R \in \mathbb{R}_0^+$ of all trials, and their maximum value at each time instant. The failure is triggered at $t^* = 0\,\mathrm{s}$ and it is evident that the position and orientation state errors increase directly after the failure but then decrease after $\approx 2.5\,\mathrm{s}$ and stabilize at small values after $\approx 4\,\mathrm{s}$. Similarly, the maximum of the state error increases in the beginning, reaches its maximum after $\approx 2.5\,\mathrm{s}$ but then decreases rapidly.

In Experiment 2-2, (see Figure 6.13) a step in the reference position $\mathbf{p}_r$ of $0.5\,\mathrm{m}$ is commanded at $t = 20\,\mathrm{s}$ under failed condition (FC). At $t = 55\,\mathrm{s}$ an opposing step of $-0.5\,\mathrm{m}$ is commanded. Tilt-Hex tracks both steps within a few seconds and the platform position and orientation remains perfectly stable.

In Experiment 2-3 the reference position trajectory is gradually changed about all axes with a total trajectory length of $2.4\,\mathrm{m}$ (see Figure 6.14, first plot) while the reference

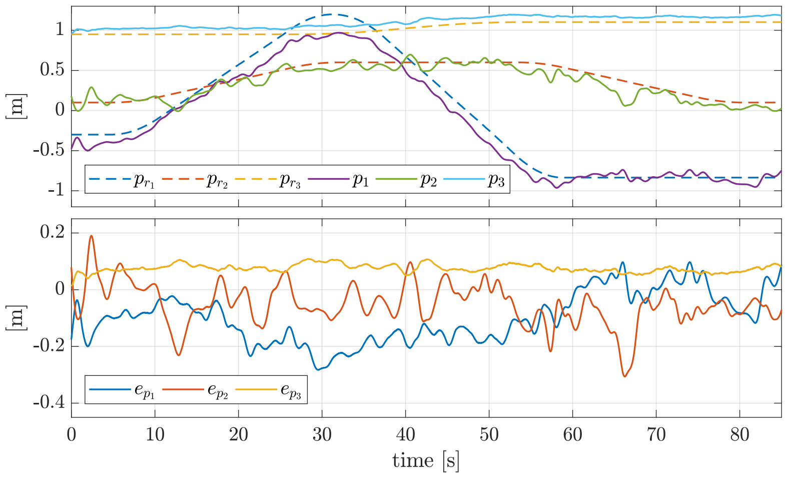


Figure 6.14. Experiment 2-3 - Trajectory following of the system with failure.

orientation is horizontal ($\mathbf{R}_r = \mathbf{I}_3$). The position error remains limited with a maximum norm of 0.3 m at 59 s. Note that failed Tilt-Hex is actually more difficult to control than an ordinary under-actuated system (e.g., a standard quadrotor). In a collinear multi-rotor system the generated thrust force is always perpendicular to the rotor plane regardless of the rotational speed of each rotor. In failed Tilt-Hex, this property is not given anymore, making the tracking of time varying trajectories much more difficult.

6.4 Results Summary

This chapter addresses the control task of stabilizing a multi-rotor platform to a given reference position with an arbitrary but constant orientation. Two control schemes are presented assuming that the given GTM can statically hover according to the Definition 5.3.5 and that its attitude is expressed using the quaternion and the rotation matrix formalism, respectively.

The first solution is a state feedback non-linear controller mainly based upon the existence of a preferential direction in the feasible force space and upon the assumption of a certain level of decoupling between the control force and control moment. Numerical simulation results show the effectiveness of the proposed control law, and the asymptotic stability of the error dynamics is proved through a cascaded structure and exploiting nested sets and reduction theorems.

The second solution is a cascaded geometric controller that results to be effective also for partially coupled platforms. In this perspective, the control law is again based on the existence of a preferential direction. This has been validated through simulative and experimental tests on hexarotor GTMs subject to the loss of a propeller, confirming also the findings of the previous chapter. In particular, the robustness of the tilted star-shaped hexarotor has been proved both by simulative and experimental results, while the capability of the Y-shaped hexarotor to realize the static hovering condition after a rotor loss has been validated through a realistic simulation.

7

Rigidity-Based Formation Control for Swarms of UAVs

Nowadays, robotics community is making significant effort to develop effective, distributed and scalable control laws for large groups of aerial platforms. The interest toward this topic is motivated by the broad potential applications of UAVs swarms in civilian and military areas, such as wide areas surveillance and contour mapping, anomalies detection, target search and localization, and so on. Many formation control strategies have been proposed in past decades, such as leader-follower, behavior, virtual structure based approaches; nevertheless a new research direction is recently emerged resting upon the exploitation of the rigidity theory.

Requiring that the bearing measurements among the agents are maintained, a rigidity-based formation control law is introduced in this chapter for an aerial multi-vehicles system. Two independent control tasks are taking into account: the stabilization and the coordinated motion of a given formation.

The contents of this chapter are available in
Michieletto G., Cenedese A., and Franchi A. Bearing rigidity theory in SE(3). *IEEE 55th Conference on Decision and Control (CDC)*, pages 5950–5955, 2016.

7.1 Formation Control Based on Rigidity Theory

Trying to mimic the emergent properties of animal flocks, a vast number of decentralized control strategies have been designed in the last ten years to stabilize a formation of mobile aerial (or also terrestrial) robots for the purpose of achieving a global objective (see e.g., Liang and Lee (2006); Turpin, Michael, and Kumar (2012); Antonelli, Arrichiello, Caccavale, and Marino (2014); Dong, Yu, Shi, and Zhong (2015); Oh, Park, and Ahn (2015) and the references therein).

Most of decentralized formation controllers aims at coordinating a multi-agent system to realize some desired spatial arrangement and/or to perform some desired motions. For this purpose, only relative measurements are employed, exploiting the local sensing capabilities of each vehicle in the group (Cao, Yu, and Anderson (2011); Liu and Jiang (2013)). Under these premises, the theory of *formation rigidity* is emerged as the most suitable theoretical framework to address this kind of control problem for multi-agent UAVs systems.

7.1.1 Literature Overview

According to the original definition of Asimow and Roth (1979), the rigidity theory aims at studying the stiffness properties of complex systems made up of different units connected among them by flexible linkages or hinges. However, this theory has recently enlarged its focus to the autonomous multi-agent systems wherein the connections are virtual, representing the sensing interplay among the devices, namely the available relative measurements. In this new perspective, the rigidity framework fits for applications connected with the control of swarms of mobile robots for localization, exploration, mapping and tracking of a target (Olfati-Saber and Murray (2002); Eren (2007); Wu, Zhang, Sheng, and Kanchi (2010); Zelazo, Franchi, Bülthoff, and Giordano (2015a)). More generally, it turns out to be an important architectural property of many multi-agent systems where a common inertial reference frame is unavailable but the agents involved are characterized by sensing, communication and movement capability.

According to the available sensing measurements, rigidity properties for a framework deal with inter-agent distance and/or direction maintenance. When agents are able to gather only range information, distance constraints can be imposed to preserve distance rigidity properties (Zelazo, Franchi, Allgöwer, Bülthoff, and Giordano (2012); Zelazo, Franchi, and Giordano (2014)). On the other hand, parallel/bearing rigidity properties are determined by direction contraints defined upon angle/bearing measurements (Eren, Whiteley, Belhumeur, Anderson, et al. (2003); Franchi and Giordano (2012); Zelazo et al. (2014); Zelazo, Giordano, and Franchi (2015b); Zhao and Zelazo (2016)). *Bearing rigidity* in particular refers to the case in which the agents are equipped with sensors providing angular/bearing measurements with regard to the neighbors: this is for instance the case of mobile robots or aerial vehicles equipped with on-board omnidirectional cameras.

In the last decade, bearing rigidity theory has been investigated taking into account agents no longer modeled as massless points in $\mathbb{R}^d$ with $d \in \mathbb{N}, d \geq 2$ (Zhao and Zelazo (2016)) but as rigid bodies. In this sense, each element of the given formation is characterized by a position and an orientation with respect to the inertial world frame, and both of them can change over time according to the agent motion capability. For example, Zelazo et al. (2014, 2015b) have coded the bearing rigidity theory for frameworks embedded in the two-dimensional Special Euclidean space $SE(2) = \mathbb{R}^2 \times \mathbb{S}^1$, accounting for teams of robots characterized by two DoFs for the translational dynamics and one DoF for the rotational one. Subsequently, Spica and Giordano (2016); Schiano, Franchi, Zelazo, and Giordano (2016) have extended the study to agents acting in 3D environment, although limiting their attitude kinematics to rotations along only one axis so that the agent domain is $\mathbb{R}^3 \times \mathbb{S}^1$. In both these scenarios, the rigidity properties of a framework are determined by evaluating the so-called *bearing rigidity matrix* which attends the measurements dynamics.

Developed notions on bearing rigidity theory have then been used to face formation control issues. Zelazo et al. (2015b) have proposed a distributed control law that exploits bearing only information to drive a given $SE(2)$ rigid formation to a configuration that is bearing congruent to the desired one. The proposed solution exploits the gradient-based minimization of a suitable potential function which accounts for the rigidity properties evolution of the framework. Likewise, a decentralized strategy has been introduced by Schiano et al. (2016) for controlling a group of quadrotors measuring relative bearings in their own body frames. Two control objectives are addressed in the $\mathbb{R}^3 \times \mathbb{S}^1$ space, namely the stabilization of the formation towards a desired configuration, and the coordinated motion along directions in the null space of its bearing rigidity matrix.

7.1.2 Problem formulation

Motivated by the current state of the art, the three-dimensional Special Euclidean space $SE(3) = \mathbb{R}^3 \times SO(3)$ is considered in this chapter, dealing with more realistic and complex situations. This manifold indeed characterizes the fully-actuated GTM vehicles introduced in the previous chapters, namely multi-rotor platforms modeled as rigid bodies having full motion capabilities in the 3D space and six controllable DoFs (Figure 5.1).

Let consider a formation of n fully-actuated GTM agents, which could be used for complex physical and human interaction tasks (Ryll et al. (2017)). The kinematic and dynamic model of each platform in the group is given by (5.1),(5.9)-(5.10). Hence $\chi_i = (\mathbf{p}_i, \mathbf{R}_i) \in SE(3)$ describes the pose of each vehicle in the inertial world frame $\mathscr{F}_W$, which is assumed to be unavailable to the group. For sake of notation compactness, the following quantities are introduced: $\mathbf{p} \in \mathbb{R}^{3n}$ and $\mathbf{R} \in SO(3)^n$ that respectively stack the positions and the attitudes of all the agents composing the formation, and $\boldsymbol{\chi} \in SE(3)^n$ that groups all the GTM poses.

It is assumed that all the agents in the formation are equipped with on-board omnidirectional calibrated cameras so that the i-th vehicle can measure its *relative bearing* with

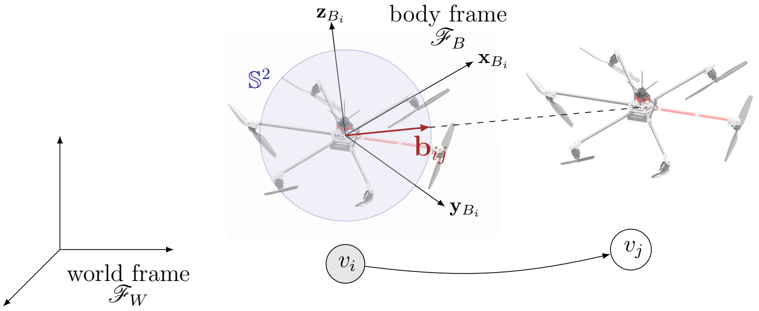


Figure 7.1. Bearing measurement for two GTM platforms belonging to a formation.

respect to the j-th vehicle (see Figure 7.1), that expressed in its local body frame is

$$\mathbf{b}_{ij} = \mathbf{R}_i^\top \frac{\mathbf{p}_j - \mathbf{p}_i}{\|\mathbf{p}_j - \mathbf{p}_i\|_2} = \mathbf{R}_i^\top \bar{\mathbf{p}}_{ij} \in \mathbb{S}^2. \tag{7.1}$$

The sensing and also communication capabilities of the whole multi-agent system are then described by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n, |\mathcal{E}| = m$ (*sensing and communication* graph). In this sense, an edge $(v_i, v_j) \in \mathcal{E}$ represents the possibility for the i-th agent both to measure its relative bearing with respect to the j-th agent, namely to recover $\mathbf{b}_{ij} \in \mathbb{S}^2$, and to communicate with it. The choice of using oriented edges is mainly motivated by real mutual visibility constraints that impose unilateral interactions.

Given these premises, a $SE(3)$ *framework* is characterized by the formation configuration, in terms of vehicles poses, and by the topology associated to agents interaction.

Definition 7.1.1 ($SE(3)$ framework)**.** A n-agents $SE(3)$ framework coincides with the pair $(\mathcal{G}, \boldsymbol{\chi})$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed graph and $\boldsymbol{\chi} \in SE(3)^n$ describes the pose of all the agents within the group.

For these frameworks, the bearing rigidity properties are investigated in this chapter, wherein also a double control task is faced, namely the stabilization of a given formation and its coordinated motion along the directions that ensure the measurement preservation.

7.2 $SE(3)$-Bearing Rigidity Theory

Bearing rigidity theory in $SE(3)$ seeks to evaluate the rigidity properties of a given $SE(3)$ framework resting on its bearing measurements preservation. In other words, the goal is to identify the motions that do not modify the systems in term of inter-agent bearings.

Given a $SE(3)$ framework $(\mathcal{G}, \boldsymbol{\chi})$, the information about the available measurements can be handled defining the $SE(3)$-*bearing function.* This is the map

$$\mathbf{b}_{\mathcal{G}} : SE(3)^n \to \mathbb{S}^{2^m} \tag{7.2}$$

$$\boldsymbol{\chi} \to \mathbf{b}_{\mathcal{G}}(\boldsymbol{\chi}) = \left[\mathbf{b}_1^\top \dots \mathbf{b}_m^\top\right]^\top \tag{7.3}$$

where $\mathbf{b}_k, k \in \{1 \dots m\}$ denotes the measurement on the k-th directed edge in $\mathcal{G}$, according to any chosen labelling on $\mathcal{E}$. It can be proven that the $SE(3)$-bearing function can be written in the compact form

$$\mathbf{b}_{\mathcal{G}}(\boldsymbol{\chi}) = -\operatorname{diag}\left(\left\{\frac{\mathbf{R}_i^\top}{\|\mathbf{p}_j - \mathbf{p}_i\|_2}\right\}\right)(\mathbf{E} \otimes \mathbf{I}_3)^\top \mathbf{p},$$

where $\mathbf{E} \in \mathbb{R}^{n \times m}$ is the incident matrix associated to the graph $\mathcal{G}$ (according to the notation given in Section 1.2.2).

Definition 7.2.1 (Bearing Equivalence and Congruence in $SE(3)$)**.** Two frameworks $(\mathcal{G}, \boldsymbol{\chi})$ and $(\mathcal{G}, \boldsymbol{\chi}')$ are said to be *bearing equivalent* (BE) if $\mathbf{b}_{ij} = \mathbf{b}'_{ij}\ \forall (v_i, v_j) \in \mathcal{E}$ and *bearing congruent* (BC) if $\mathbf{b}_{ij} = \mathbf{b}'_{ij}\ \forall (v_i, v_j) \in \mathcal{V} \times \mathcal{V}$.

Roughly speaking, equivalence deals with the preservation of the measurements between interacting agents, whereas congruence implies that bearings are maintained among all the possible pairs of nodes in the group. These two properties coincide when the sensing (and communication) capabilities are represented by the complete directed graph $\mathcal{K}$ having $m = n(n-1)$ edges: it represents the case wherein all the agents are connected, hence all the mutual bearing measurements are available.

Accounting for the preimage[1] under the $SE(3)$-bearing function, the set $\mathcal{C}(\boldsymbol{\chi}) = \mathbf{b}_{\mathcal{K}}^{-1}(\mathbf{b}_{\mathcal{K}}(\boldsymbol{\chi})) \subset SE(3)^n$ contains the configuration $\boldsymbol{\chi}$ and all its possible transformations induced by the complete graph $\mathcal{K}$, namely all the $\boldsymbol{\chi}'$ such that $(\mathcal{G}, \boldsymbol{\chi}')$ is BC to $(\mathcal{G}, \boldsymbol{\chi})$. On the other hand, the set $\mathcal{Q}(\boldsymbol{\chi}) = \mathbf{b}_{\mathcal{G}}^{-1}(\mathbf{b}_{\mathcal{G}}(\boldsymbol{\chi})) \subset SE(3)^n$ contains (at least) $\boldsymbol{\chi}$ and all its possible transformations induced by the graph $\mathcal{G}$, i.e., all the $\boldsymbol{\chi}'$ such that $(\mathcal{G}, \boldsymbol{\chi}')$ is BE to $(\mathcal{G}, \boldsymbol{\chi})$. Trivially, it holds that $\{\boldsymbol{\chi}\} \subseteq \mathcal{C}(\boldsymbol{\chi}) \subseteq \mathcal{Q}(\boldsymbol{\chi})$.

In such a scenario, a framework $(\mathcal{G}, \boldsymbol{\chi})$ is $SE(3)$-bearing rigid when for any $\boldsymbol{\chi}' \in SE(3)^n$ sufficiently close to $\boldsymbol{\chi}$ with the same bearing measurements, namely $\boldsymbol{\chi} \in \mathcal{Q}(\boldsymbol{\chi})$, there exists a (local) bearing-preserving transformation taking $\boldsymbol{\chi}$ to $\boldsymbol{\chi}'$.

Definition 7.2.2 (Bearing Rigidity in $SE(3)$)**.** A framework $(\mathcal{G}, \boldsymbol{\chi})$ is said to be *SE(3)-bearing rigid* (BR) if there exists a neighbor $\mathcal{S}(\boldsymbol{\chi})$ of $\boldsymbol{\chi} \in SE(3)^n$ such that

$$\mathcal{C}(\boldsymbol{\chi}) \cap \mathcal{S}(\boldsymbol{\chi}) = \mathcal{Q}(\boldsymbol{\chi}) \cap \mathcal{S}(\boldsymbol{\chi}). \tag{7.4}$$

In other words, there exists a neighbor $\mathcal{S}(\boldsymbol{\chi})$ of the current configuration $\boldsymbol{\chi}$ such that each framework $(\mathcal{G}, \boldsymbol{\chi}'), \boldsymbol{\chi}' \in \mathcal{S}(\boldsymbol{\chi})$ that is BE is also BC to $(\mathcal{G}, \boldsymbol{\chi})$. Figure 7.2 aims at graphically clarifying the condition (7.4).

Definition 7.2.3 (Global Bearing Rigidity in $SE(3)$)**.** A framework $(\mathcal{G}, \boldsymbol{\chi})$ is said to be *globally $SE(3)$-bearing rigid* (GBR) if every framework that is BE to $(\mathcal{G}, \boldsymbol{\chi})$ is also BC.

GBR property can be reformulated in terms of bearing rigidity over $SE(3)$, namely assuming that $\mathcal{S}(\boldsymbol{\chi}) = SE(3)$. Therefore, condition (7.4) boils down to the equivalence

[1] Let $f\colon \mathcal{X} \rightarrow \mathcal{Y}$ be a function. Let $\mathcal{A} \subset \mathcal{X}$ and $\mathcal{B} \subset \mathcal{Y}$. Then $f(\mathcal{A}) = \{f(x) \in \mathcal{Y} \mid x \in \mathcal{A}\}$ is called the *image* of $\mathcal{A}$ under f and $f^{-1}(\mathcal{B}) = \{x \in \mathcal{X} \mid f(x) \in \mathcal{B}\}$ is called the *preimage* of $\mathcal{B}$ under f.

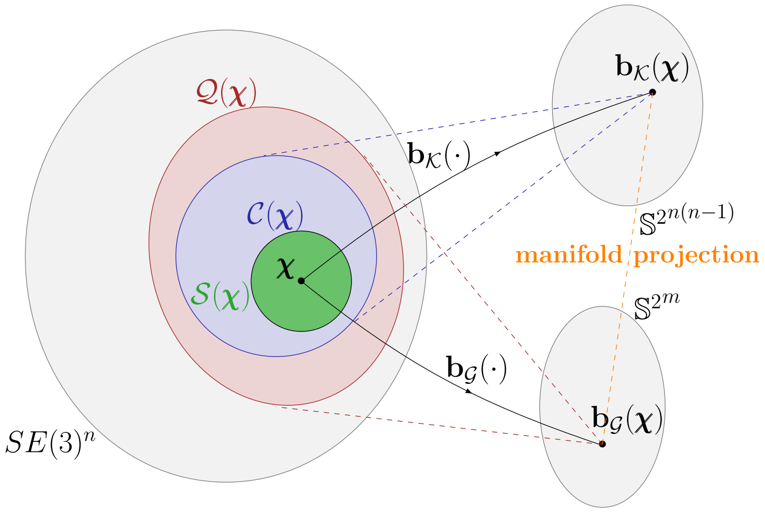

$\mathbf{b}_{\mathcal{K}}(\boldsymbol{\chi})$
$\mathcal{Q}(\boldsymbol{\chi})$
$\mathbf{b}_{\mathcal{K}}(\cdot)$
$\mathcal{C}(\boldsymbol{\chi})$
$\mathbb{S}^{2n(n-1)}$
$\boldsymbol{\chi}$
manifold projection
$\mathcal{S}(\boldsymbol{\chi})$
$\mathbb{S}^{2m}$
$\mathbf{b}_{\mathcal{G}}(\cdot)$
$\mathbf{b}_{\mathcal{G}}(\boldsymbol{\chi})$
$SE(3)^n$

Figure 7.2. Graphical interpretation of condition (7.4): $\exists \mathcal{S}(\boldsymbol{\chi}) \in SE(3)^n$ such that $\mathcal{C}(\boldsymbol{\chi}) \cap \mathcal{S}(\boldsymbol{\chi}) = \mathcal{Q}(\boldsymbol{\chi}) \cap \mathcal{S}(\boldsymbol{\chi})$.

$\mathcal{C}(\boldsymbol{\chi}) = \mathcal{Q}(\boldsymbol{\chi})$ (imposing that $\mathcal{C}(\boldsymbol{\chi})$ is not a singleton, i.e., $\{\boldsymbol{\chi}\} \subsetneq \mathcal{C}(\boldsymbol{\chi})$). Hence, it is straightforward that global bearing rigidity implies bearing rigidity but not viceversa.

Let introduce now the difference set $\mathbf{b}_{\mathcal{G}}^{-1}(\mathbf{b}_{\mathcal{G}}(\boldsymbol{\chi}))\backslash\mathbf{b}_{\mathcal{K}}^{-1}(\mathbf{b}_{\mathcal{K}}(\boldsymbol{\chi}))$ (or equivalently $\mathcal{Q}(\boldsymbol{\chi})\backslash\mathcal{C}(\boldsymbol{\chi})$) which contains all the possible transformations of $\boldsymbol{\chi}$ constrained by $\mathcal{G}$ that are not admissible by $\mathcal{K}$. This allows to define the property of roto-flexibility for a framework.

Definition 7.2.4 (Bearing Roto-Flexibility in $SE(3)$)**.** A framework $(\mathcal{G}, \boldsymbol{\chi})$ is said to be $SE(3)$-bearing roto-flexible (BRF) if there exists an analytic path $T\colon [0,1] \to SE(3)^n$ such that

$$T(0) = \boldsymbol{\chi} \quad \text{and} \tag{7.5}$$

$$T(t) \in \mathbf{b}_{\mathcal{G}}^{-1}(\mathbf{b}_{\mathcal{G}}(\boldsymbol{\chi}))\backslash\mathbf{b}_{\mathcal{K}}^{-1}(\mathbf{b}_{\mathcal{K}}(\boldsymbol{\chi}))\ \forall t \in (0,1]. \tag{7.6}$$

Note that a framework $(\mathcal{G}, \boldsymbol{\chi})$ is BRF if $\mathcal{C}(\boldsymbol{\chi})$ is a singleton, namely $\{\boldsymbol{\chi}\} = \mathcal{C}(\boldsymbol{\chi})$, while the difference set $\mathcal{Q}(\boldsymbol{\chi})\backslash\mathcal{C}(\boldsymbol{\chi})$ is not, hence it holds that $\mathcal{C}(\boldsymbol{\chi}) \subsetneq \mathcal{Q}(\boldsymbol{\chi})$. By using the set theory, it will be proved that a BRF framework is not infinitesimal $SE(3)$-bearing rigid.

7.2.1 Infinitesimal Rigidity Properties

For a given framework $(\mathcal{G}, \boldsymbol{\chi})$, the property of *infinitesimal* $SE(3)$*-bearing rigidity* is related to the conditions that, within a dynamic context, ensure measurements maintenance among all the pairs of agents in the formation.

Let introduce the $SE(3)$*-bearing rigidity matrix* that associates the variations of the available measurements to the agents dynamics, namely the matrix $\mathbf{B}_{\mathcal{G}}(\boldsymbol{\chi}) \in \mathbb{R}^{3m\times 6n}$ such that

$$\dot{\mathbf{b}}_{\mathcal{G}}(\boldsymbol{\chi}) = \mathbf{B}_{\mathcal{G}}(\boldsymbol{\chi}) \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}, \tag{7.7}$$

$$\begin{tikzcd}
\begin{array}{c}\text{GBR}\\ \{\chi\} \subsetneq \mathcal{C}(\chi) = \mathcal{Q}(\chi)\end{array} \arrow[r, Leftrightarrow] \arrow[d, Rightarrow] & \begin{array}{c}\text{IBR}\\ \{\chi\} \subsetneq \mathcal{C}(\chi) = \mathcal{Q}(\chi)\end{array} \arrow[d, Leftrightarrow] \\
\begin{array}{c}\text{BR}\\ \{\chi\} \subsetneq \mathcal{C}(\chi) \subseteq \mathcal{Q}(\chi)\end{array} & \begin{array}{c}\neg\text{BRF}\\ \{\chi\} = \mathcal{C}(\chi) \subsetneq \mathcal{Q}(\chi)\end{array}
\end{tikzcd}$$

Figure 7.3. Relations between the main $SE(3)$-bearing rigidity concepts, namely bearing rigidity (BR), global bearing rigidity (GBR), infinitesimal bearing rigidity (IBR), bearing roto-flexibility (BRF).

where $[\mathbf{v}^\top \; \boldsymbol{\omega}^\top]^\top \in \mathbb{R}^{6n}$ is the vector that stacks the linear and angular velocities of all the agents expressed in their body frame. It can be proved that the k-th row block of $\mathbf{B}_\mathcal{G}(\boldsymbol{\chi})$ (corresponding to $\mathbf{b}_k = \mathbf{b}_{ij}$) results to be

$$\begin{array}{ccccccc} & i & & j & & 3n+i & \\ \Big[\mathbf{0} & -\frac{P(\mathbf{b}_{ij})}{\|\mathbf{p}_j - \mathbf{p}_i\|_2} & \mathbf{0} & \frac{P(\mathbf{b}_{ij})}{\|\mathbf{p}_j - \mathbf{p}_i\|_2}\mathbf{R}_i^\top \mathbf{R}_j & \mathbf{0} & [\mathbf{b}_{ij}]_\times & \mathbf{0}\Big] \end{array} \tag{7.8}$$

where $P(\mathbf{b}_{ij}) = \mathbf{I}_3 - \mathbf{b}_{ij}\mathbf{b}_{ij}^\top \in \mathbb{R}^{3\times 3}$ is the orthogonal projector of $\mathbf{b}_{ij}$ (see Remark 6.3.2).

Definition 7.2.5 (Infinitesimal and Trivial Motions in $SE(3)$)**.** An *infinitesimal motion* of a $SE(3)$ framework $(\mathcal{G}, \boldsymbol{\chi})$ is a smooth motion along a path $T\colon [0,1] \to SE(3)^n$ with $T(0) = \boldsymbol{\chi}$ such that the time derivative of the associated rigidity function is zero. If the $SE(3)$ framework is $(\mathcal{K}, \boldsymbol{\chi})$ then the infinitesimal motion is called *trivial motion.*

As a consequence of Definition 7.2.5, if a path T is contained entirely in $\mathbf{b}_\mathcal{G}^{-1}(\mathbf{b}_\mathcal{G}(\boldsymbol{\chi}))$ (i.e., $\mathcal{Q}(\boldsymbol{\chi})$) for all $t \in [0,1]$, then the infinitesimal motions are completely described by the tangent bundle[2] $\mathfrak{T}_\mathcal{Q}$ of $\mathbf{b}_\mathcal{G}^{-1}(\mathbf{b}_\mathcal{G}(\boldsymbol{\chi}))$ and it can be proved that $\mathfrak{T}_\mathcal{Q} = \ker(\mathbf{B}_\mathcal{G}(\boldsymbol{\chi}))$. On the other hand, the trivial motions are identified by the tangent bundle $\mathfrak{T}_\mathcal{C}$ of $\mathbf{b}_\mathcal{K}^{-1}(\mathbf{b}_\mathcal{K}(\boldsymbol{\chi}))$ which is so that $\mathfrak{T}_\mathcal{C} = \ker(\mathbf{B}_\mathcal{K}(\boldsymbol{\chi})) \subseteq \ker(\mathbf{B}_\mathcal{G}(\boldsymbol{\chi}))$. These premises lead to a formal definition for infinitesimal bearing rigidity of frameworks in $SE(3)$.

Definition 7.2.6 (Infinitesimal Bearing Rigidity in $SE(3)$)**.** A framework $(\mathcal{G}, \boldsymbol{\chi})$ is said to be *infinitesimally $SE(3)$-bearing rigid* (IBR) if $\ker(\mathbf{B}_\mathcal{G}(\boldsymbol{\chi})) = \ker(\mathbf{B}_\mathcal{K}(\boldsymbol{\chi}))$.

The infinitesimal bearing rigidity deals with the equivalence between trivial and infinitesimal motions. Because infinitesimal motions are associated to paths in $\mathcal{Q}(\boldsymbol{\chi})$ while trivial motion are related to paths in $\mathcal{C}(\boldsymbol{\chi})$, a framework $(\mathcal{G}, \boldsymbol{\chi})$ is IBR if $\{\boldsymbol{\chi}\} \subsetneq \mathcal{C}(\boldsymbol{\chi}) = \mathcal{Q}(\boldsymbol{\chi})$. Note that the last condition implies that an infinitesimally rigid framework is not roto-flexible while it is globally rigid, and viceversa (except for pathological cases). All these implications are graphically summarized in Figure 7.3.

In general, infinitesimal motions of a $SE(3)$ framework arise from the composition of motions in $\mathbb{R}^3$ with motions in $SO(3)$ for each agent. Each infinitesimal motion $\boldsymbol{\delta} \in \ker(\mathbf{B}_\mathcal{G}(\boldsymbol{\chi}))$ of the formation is therefore characterized by a linear velocity component $\boldsymbol{\delta}_v \in \mathbb{R}^{3n}$ and a angular velocity component $\boldsymbol{\delta}_\omega \in \mathbb{R}^{3n}$.

[2] The *tangent bundle* of a differentiable manifold $\mathcal{M}$ is a manifold $\mathfrak{T}_\mathcal{M}$, which assembles all the tangent vectors in $\mathcal{M}$, namely $\mathfrak{T}_\mathcal{M} = \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in \mathcal{M}, \mathbf{y} \in \mathfrak{T}_\mathbf{x}\}$ where $\mathfrak{T}_\mathbf{x}$ denotes the tangent space to $\mathcal{M}$ at the point $\mathbf{x}$.

Proposition 7.2.7. *Every infinitesimal motion* $\boldsymbol{\delta}_\chi \in \ker(\mathbf{B}_\mathcal{G}(\chi))$ *satisfies the condition*

$$\left(\mathbf{B}_{\|,\mathcal{G}}(\mathbf{p})\,\mathrm{diag}\left(\{\mathbf{R}_i\}\right)\right)\,\boldsymbol{\delta}_v = \left(\mathrm{diag}(\{-\mathbf{R}_i[\mathbf{b}_{ij}]_\times\})(\mathbf{E}_\mathrm{o}\otimes\mathbf{I}_3)^\top\right)\,\boldsymbol{\delta}_\omega, \tag{7.9}$$

where

- $\mathbf{B}_{\|,\mathcal{G}}(\mathbf{p}) = \mathrm{diag}\left(\left\{\frac{P(\bar{\mathbf{p}}_{ij})}{\|\mathbf{p}_j-\mathbf{p}_i\|_2}\right\}\right)(\mathbf{E}\otimes\mathbf{I}_3)^\top$ *is the bearing rigidity matrix in* $\mathbb{R}^3$ *(see Zhao and Zelazo (2016)),*
- *the* $\{0,1\}$*-matrix* $\mathbf{E}_\mathrm{o}\in\mathbb{R}^{n\times n}$ *is designed so that* $[\mathbf{E}_\mathrm{o}]_{ik}=1$ *if* $e_k=(v_i,v_j)\in\mathcal{E}$ *(outgoing edge) and* $[\mathbf{E}_\mathrm{o}]_{ik}=0$ *otherwise.*

Proof. The proof derives from the expression (7.8), observing that it holds

$$\mathbf{R}_i^\top\frac{P(\bar{\mathbf{p}}_{ij})}{\|\mathbf{p}_j-\mathbf{p}_i\|_2} = \frac{P(\mathbf{b}_{ij})}{\|\mathbf{p}_j-\mathbf{p}_i\|_2}\mathbf{R}_i^\top. \tag{7.10}$$

Exploiting (7.10), it is straightforward that the bearing rigidity matrix results as follows

$$\mathbf{B}_\mathcal{G}(\boldsymbol{\chi}) = \left[\mathrm{diag}\left(\left\{\mathbf{R}_i^\top\frac{P(\bar{\mathbf{p}}_{ij})}{\|\mathbf{p}_j-\mathbf{p}_i\|_2}\right\}\right)(\mathbf{E}\otimes\mathbf{I}_3)^\top\,\mathrm{diag}\left(\{\mathbf{R}_i\}\right)\quad \mathrm{diag}(\{[\mathbf{b}_{ij}]_\times\})(\mathbf{E}_\mathrm{o}\otimes\mathbf{I}_3)^\top\right], \tag{7.11}$$

so, doing suitable computations, the thesis is proved. □

Proposition 7.2.7 infers a relationship between the infinitesimal motions of a $SE(3)$ framework and those of a infinitesimal bearing rigid framework in $\mathbb{R}^3$. Indeed, if all the agents do not alter their attitude, i.e., $\boldsymbol{\delta}_\omega = 0$, then the condition $\boldsymbol{\delta}_\chi\in\ker(\mathbf{B}_\mathcal{G}(\boldsymbol{\chi}))$ reduces to the condition to evaluate infinitesimal bearing rigidity in $\mathbb{R}^3$ (Zhao and Zelazo (2016)). Hence, translations and uniform scaling of the framework result to be trivial motions still in $SE(3)$ context. On the contrary, if the angular velocities of the agents are non-zero, then the trivial motions of the framework consist of the bearing-preserving motions termed *3D coordinated rotations*. They envisage a 3D synchronized rotation of all the agents in their frameworks jointly to an equal rotation of the whole framework as a rigid body in $\mathbb{R}^3$ (the angular velocity of each agent is equal to the one of the formation CoM). In the following, the space of the 3D coordinated rotation associated to the graph $\mathcal{G}$ is denoted as $\mathcal{R}_\circlearrowleft(\mathcal{G})$. Figure 7.4 proposes an explanatory example of the just described motions.

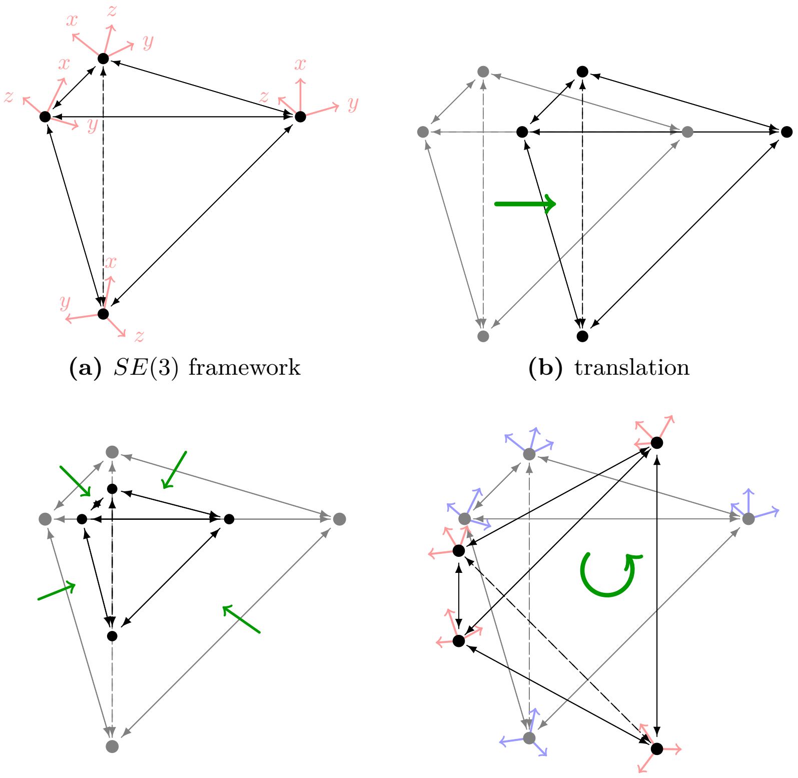
Proposition 7.2.8. *In general, the coordinated rotation pace* $\mathcal{R}_\circlearrowleft(\mathcal{G})$ *is not trivial, i.e.,* $dim\mathcal{R}_\circlearrowleft(\mathcal{G})\geq 1$.

Proof. Consider a rigid body rotation of the whole framework $(\mathcal{G},\boldsymbol{\chi})$ generated by the angular velocity $\boldsymbol{\omega}'$, jointly with the rotation of each agent with the same angular velocity. Assume that $\boldsymbol{\omega}'$ is expressed in world reference frame and is arbitrarily taken from $\mathbb{R}^3$.

The described motion is characterized by the two following components

$$\boldsymbol{\delta}_v = -\,\mathrm{diag}(\{\mathbf{R}_i^\top[\mathbf{p}_i]_\times\})(\mathbb{1}_n\otimes\boldsymbol{\omega}') \tag{7.12}$$

$$\boldsymbol{\delta}_\omega = \mathrm{diag}(\{\mathbf{R}_i^\top\})(\mathbb{1}_n\otimes\boldsymbol{\omega}') \tag{7.13}$$



(a) $SE(3)$ framework

(b) translation

(c) uniform scaling

(d) 3D coordinated rotation

Figure 7.4. Example of trivial motions: (a) initial 4-agents $SE(3)$ framework $(\mathcal{K}, \boldsymbol{\chi})$, (b) translation along a single axis, (c) scaling by a factor $\xi = 0.5$, (d) counterclockwise coordinated rotation of $\theta = \pi/4$.

Applying to vectors (7.12)-(7.13) the condition stated in Proposition 7.2.7 for the infinitesimal motions, each equivalence of the system (7.9) results to be valid. Indeed it holds that

$$-\frac{P(\bar{\mathbf{p}}_{ij})}{\|\mathbf{p}_j - \mathbf{p}_i\|_2}([\mathbf{p}_j]_\times - [\mathbf{p}_i]_\times)\,\boldsymbol{\omega}' = -\mathbf{R}_i[\mathbf{b}_{ij}]_\times \mathbf{R}_i^\top\,\boldsymbol{\omega}' \tag{7.14}$$

$$-P(\bar{\mathbf{p}}_{ij})[\bar{\mathbf{p}}_{ij}]_\times\,\boldsymbol{\omega}' = -[\mathbf{R}_i\mathbf{b}_{ij}]_\times\,\boldsymbol{\omega}' \tag{7.15}$$

$$-(\mathbf{I}_3 - \bar{\mathbf{p}}_{ij}\bar{\mathbf{p}}_{ij}^\top)[\bar{\mathbf{p}}_{ij}]_\times\,\boldsymbol{\omega}' = -[\bar{\mathbf{p}}_{ij}]_\times\,\boldsymbol{\omega}' \tag{7.16}$$

$$-[\bar{\mathbf{p}}_{ij}]_\times\,\boldsymbol{\omega}' = -[\bar{\mathbf{p}}_{ij}]_\times\,\boldsymbol{\omega}' \tag{7.17}$$

The described motion is thus an infinitesimal motion for any $\boldsymbol{\omega}' \in \mathbb{R}^3$ and it corresponds to a 3D coordinated rotation. Hence, in the space $\mathcal{R}_\circlearrowleft(\mathcal{G})$ there is at least a vector. □

Corollary 7.2.9. *For a complete directed graph $\mathcal{K}$, the coordinated rotation space is such that $dim\mathcal{R}_\circlearrowleft(\mathcal{K}) = 3$.*

Proof. When the sensing and communication capabilities of the multi-agent formation are described by the complete directed graph $\mathcal{K}$, the angular velocity $\boldsymbol{\omega}'$ introduced in the previous proof can always be expressed as a linear combination of $\mathbf{e}_1 = [1\ 0\ 0]^\top$, $\mathbf{e}_2 = [0\ 1\ 0]^\top$, $\mathbf{e}_3 = [0\ 0\ 1]^\top$. Then the space $\mathcal{R}_\circlearrowleft(\mathcal{G})$ has dimension equal to 3. □

Corollary 7.2.10. *For an infinitesimally rigid $SE(3)$ framework $(\mathcal{G}, \boldsymbol{\chi})$, it holds that*

$$\ker(\mathbf{B}_{\mathcal{G}}(\boldsymbol{\chi})) = \ker(\mathbf{B}_{\mathcal{K}}(\boldsymbol{\chi})) = \{\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3\} \tag{7.18}$$

with

$$\mathbf{n}_1 = \begin{bmatrix} \mathrm{diag}(\{\mathbf{R}_i^\top\})(\mathbf{1}_n \otimes \mathbf{I}_3) \\ \mathbf{0} \end{bmatrix} \quad \mathbf{n}_2 = \begin{bmatrix} \mathrm{diag}(\{\mathbf{R}_i^\top\})\tilde{\mathbf{p}} \\ \mathbf{0} \end{bmatrix}$$
$$\mathbf{n}_3 = \begin{bmatrix} \mathrm{diag}(\{\mathbf{R}_i^\top\}) \begin{bmatrix} \mathbf{E}_1\tilde{\mathbf{p}} & \mathbf{E}_2\tilde{\mathbf{p}} & \mathbf{E}_3\tilde{\mathbf{p}} \end{bmatrix} \\ \mathrm{diag}(\{\mathbf{R}_i^\top\})(\mathbf{1}_n \otimes \mathbf{I}_3) \end{bmatrix} \tag{7.19}$$

where $\mathbf{E}_i = (\mathbf{I}_n \otimes [\mathbf{e}_i]_\times)$ *being* $\mathbf{e}_i$ *the* i*-th versor in* $\mathbb{R}^3$ *with* $i = 1, 2, 3$, $\mathbf{p}^* = \frac{1}{n}(\mathbf{1}_n \otimes \mathbf{I}_3)^\top \mathbf{p} \in \mathbb{R}^3$ *is the barycenter of the formation and* $\tilde{\mathbf{p}} = \mathbf{p} - (\mathbf{1}_n \otimes \mathbf{I}_3)\mathbf{p}^* \in \mathbb{R}^{3n}$.

The terms in (7.19) describe the trivial motions, namely the translations of the framework ($\mathbf{n}_1$), the uniform scaling with respect to its barycenter ($\mathbf{n}_2$) and the 3D coordinated rotation about a generic axis passing through $\mathbf{p}^*$ ($\mathbf{n}_3$). Consequence of Corollary 7.2.10 is the following theorem which constitutes the main result about the rigidity analysis for frameworks embedded in $SE(3)$.

Theorem 7.2.11. *A $SE(3)$ framework is infinitesimally rigid if and only if*

$$\mathrm{rk}(\mathbf{B}_{\mathcal{G}}(\boldsymbol{\chi})) = 6n - 7. \tag{7.20}$$

Corollary 7.2.12. *A $SE(3)$ framework is infinitesimally rigid in $SE(3)$ if and only if*

1. $\mathrm{rk}(\mathbf{B}_{\|,\mathcal{G}}(\mathbf{p})) = 3n - 4$,
2. $dim\mathcal{R}_{\circlearrowleft}(\mathcal{G}) = 3$.

7.3 Quaternion-Based Formulation

When the agents attitude is represented by a rotation matrix the orientation kinematics results to be governed by the non-linear relation (5.1). This fact implies the need of designing a highly non-linear and complex controller to govern the formation behavoir. To overcome this issue, in the following the unit quaternion representation (introduced for the GTM platforms in Section 6.2.1) is adopted for the robots orientation and the infinitesimal $SE(3)$-bearing rigidity properties are redefined using this formalism.

7.3.1 Modelling of Agents Motion and Sensing Capabilities

Let represent the attitude of the i-th robot belonging to a given n-agent formation through the unit quaternion $\mathbf{q}_i \in \mathbb{S}^3$. The $7n$-dimensional vector $\boldsymbol{\chi}^+ = \begin{bmatrix} \mathbf{p}^\top & \mathbf{q}^\top \end{bmatrix}^\top \in \mathbb{R}^{3n} \times \mathbb{S}^{3n}$, where $\mathbf{q} = \begin{bmatrix} \mathbf{q}_1^\top & \dots & \mathbf{q}_n^\top \end{bmatrix}^\top$, describes thus the framework configuration.

Exploiting the quaternion convention and assuming that the angular velocity is expressed in body frame, the orientation kinematics of any i-th agent is given by the *linear relation* (6.3),

$$\dot{\mathbf{q}}_i = \frac{1}{2}\mathbf{q}_i \odot \begin{bmatrix} 0 \\ \boldsymbol{\omega}_i \end{bmatrix} = \frac{1}{2}\mathbf{V}(\mathbf{q}_i) \begin{bmatrix} 0 \\ \boldsymbol{\omega}_i \end{bmatrix}. \tag{7.21}$$

On the other hand, recalling that each unit quaternion $\mathbf{q}_i$ is univocally associated to a rotation matrix $\mathbf{R}(\mathbf{q}_i)$ (see Eq. (B.16)), its translation kinematics results to be

$$\dot{\mathbf{p}}_i = \mathbf{R}(\mathbf{q}_i)\mathbf{v}_i. \tag{7.22}$$

Analogously, the bearing $\mathbf{b}_{ij}$ sensed by agent v_i with respect to v_j can be rewritten as $\mathbf{b}_{ij} = \mathbf{R}^\top(\mathbf{q}_i)\bar{\mathbf{p}}_{ij}$, although, because of dimension matter, hereafter the following notation will be used[3]

$$\mathbf{b}_{ij}^+ = \begin{bmatrix} 0 \\ \mathbf{b}_{ij} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{R}^\top(\mathbf{q}_i)\bar{\mathbf{p}}_{ij} \end{bmatrix} \in \mathbb{S}^3. \tag{7.23}$$

Considering the rotation of a vector using the quaternion-based attitude representation, it can be verified that

$$\mathbf{b}_{ij}^+ = \mathbf{q}_i^{-1} \odot \bar{\mathbf{p}}_{ij}^+ \odot \mathbf{q}_i = \mathbf{V}(\mathbf{q}_i^{-1})\mathbf{W}(\mathbf{q}_i)\bar{\mathbf{p}}_{ij}^+, \tag{7.24}$$

where $\bar{\mathbf{p}}_{ij}^+$ denotes the vector $[0 \quad \bar{\mathbf{p}}_{ij}^\top]^\top \in \mathbb{R}^+$, while $\mathbf{q}_i^{-1} \in \mathbb{S}^3$ is the inverse unit quaternion and the quaternion-composition rule (B.18) has been exploited.

As a consequence of expression (7.23), in the new formulation of rigidity theory the domain of the image of the bearing function (7.2) turns out to be the four-dimensional sphere at the power of m, namely $\mathbb{S}^{3m}$.

7.3.2 Infinitesimal Bearing Rigidity

To compute the bearing rigidity matrix using the quaterions formalism, the time derivative of a single bearing measurement $\mathbf{b}_{ij}^+$ has to be computed. This results

$$\dot{\mathbf{b}}_{ij}^+ = \dot{\mathbf{q}}_i^{-1} \odot \bar{\mathbf{p}}_{ij}^+ \odot \mathbf{q}_i + \mathbf{q}_i^{-1} \odot \bar{\mathbf{p}}_{ij}^+ \odot \dot{\mathbf{q}}_i + \mathbf{q}_i^{-1} \odot \dot{\bar{\mathbf{p}}}_{ij}^+ \odot \mathbf{q}_i, \tag{7.25}$$

Using the symbol $\boldsymbol{\omega}_i^+$ to identify the vector $[0 \quad \boldsymbol{\omega}_i^\top]^\top \in \mathbb{R}^4$, for the first two addends in the sum (7.25) it occurs that

$$\dot{\mathbf{q}}_i^{-1} \odot \bar{\mathbf{p}}_{ij}^+ \odot \mathbf{q}_i = -\frac{1}{2}\boldsymbol{\omega}_i^+ \odot \mathbf{q}_i^{-1} \odot \bar{\mathbf{p}}_{ij}^+ \odot \mathbf{q}_i = -\frac{1}{2}\boldsymbol{\omega}^+ \odot \mathbf{b}_{ij}^+ = -\frac{1}{2}\mathbf{W}(\mathbf{b}_{ij}^+)\boldsymbol{\omega}_i^+, \tag{7.26}$$

$$\mathbf{q}_i^{-1} \odot \bar{\mathbf{p}}_{ij}^+ \odot \dot{\mathbf{q}}_i = \frac{1}{2}\mathbf{q}_i^{-1} \odot \bar{\mathbf{p}}_{ij}^+ \odot \mathbf{q}_i \odot \boldsymbol{\omega}_i^+ = \frac{1}{2}\mathbf{b}_{ij}^+ \odot \boldsymbol{\omega}_i^+ = \frac{1}{2}\mathbf{V}(\mathbf{b}_{ij}^+)\boldsymbol{\omega}_i^+. \tag{7.27}$$

[3] Generally, in the rest of the chapter, the superscript + is used to indicated the four dimensional vector $\mathbf{x}^+ \in \mathbb{R}^4$ obtained setting a scalar zero component at the beginning of $\mathbf{x} \in \mathbb{R}^3$.

In addition, to complete the analysis, it follows that

$$\mathbf{q}_i^{-1} \odot \dot{\bar{\mathbf{p}}}_{ij}^{+} \odot \mathbf{q}_i = \begin{bmatrix} 0 \\ \mathbf{R}(\mathbf{q}_i)^\top \dot{\bar{\mathbf{p}}}_{ij} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{P(\mathbf{b}_{ij})}{\|\mathbf{p}_j - \mathbf{p}_i\|_2} \end{bmatrix} (\mathbf{R}(\mathbf{q}_i)^\top \mathbf{R}(\mathbf{q}_j)\mathbf{v}_j - \mathbf{v}_i). \tag{7.28}$$

As a consequence, through the rigidity matrix $\mathbf{B}_{\mathcal{G}}^{+}(\boldsymbol{\chi}^{+}) \in \mathbb{R}^{4m \times 7n}$, the dynamics of the bearing measurements can be expressed as function of $[\mathbf{v}^\top \;\; (\boldsymbol{\omega}^{+})^\top]^\top \in \mathbb{R}^{7n}$, where $\boldsymbol{\omega}^{+}$ stacks the angular velocity vectors $\boldsymbol{\omega}_i^{+}$ for $i = 1 \dots n$ (note the correspondence with relation (7.7)). Because of (7.26), (7.27) and (7.28), the k-th row block of $\mathbf{B}_{\mathcal{G}}^{+}(\boldsymbol{\chi})$ corresponding to $\mathbf{b}_k^{+} = \mathbf{b}_{ij}^{+}$ is such that

$$\left[\mathbf{B}_{\mathcal{G}}^{+}(\boldsymbol{\chi}^{+})\right]_{k,i} = - \begin{bmatrix} 0 \\ \frac{P(\mathbf{b}_{ij})}{\|\mathbf{p}_j - \mathbf{p}_i\|_2} \end{bmatrix} \in \mathbb{R}^{4 \times 3} \tag{7.29}$$

$$\left[\mathbf{B}_{\mathcal{G}}^{+}(\boldsymbol{\chi}^{+})\right]_{k,j} = \begin{bmatrix} 0 \\ \frac{P(\mathbf{b}_{ij})}{\|\mathbf{p}_j - \mathbf{p}_i\|_2} \end{bmatrix} \mathbf{R}(\mathbf{q}_i)^\top \mathbf{R}(\mathbf{q}_j) \in \mathbb{R}^{4 \times 3} \tag{7.30}$$

$$\left[\mathbf{B}_{\mathcal{G}}^{+}(\boldsymbol{\chi}^{+})\right]_{k,3n+i} = \mathbf{V}(\mathbf{b}_{ij}^{+}) - \mathbf{W}(\mathbf{b}_{ij}^{+}) = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & [\mathbf{b}_{ij}]_\times \end{bmatrix} \in \mathbb{R}^{4 \times 4} \tag{7.31}$$

The necessary and sufficient condition for the infinitesimal bearing rigidity is invariant as regards to the rotation representation even if the dimensions of the bearing rigidity matrix vary accordingly. It is still valid that a framework $(\mathcal{G}, \boldsymbol{\chi}^{+})$ is IBR if and only if

$$\ker(\mathbf{B}_{\mathcal{G}}^{+}(\boldsymbol{\chi}^{+})) = \ker(\mathbf{B}_{\mathcal{K}}^{+}(\boldsymbol{\chi}^{+})), \qquad \text{or equivalently,} \tag{7.32}$$

$$\mathrm{rk}(\mathbf{B}_{\mathcal{G}}^{+}(\boldsymbol{\chi}^{+})) = 6n - 7. \tag{7.33}$$

This fact could be explained considering that the total number of DoFs of the formation remain $6n$ (namely six DoFs for each of the n agents) although the parameters used to describe the agent attitude are four and not three. At the same time, the infinitesimal trivial rigid motions are still the translations and the uniform scaling of the whole framework and the 3D coordinated rotations.

Table 7.1 aims at resuming the main concepts about $SE(3)$-bearing rigidity comparing the quaternion-based formulation with the one based on the rotation matrices.

7.4 Bearing-Based Formation Control

In this section, two independent control tasks are required to be solved through a decentralized approach. Given a multi-agent formation, the attention is first focused on the *formation stabilization* proposing a distributed law that brings a given rigid framework into a desired configuration defined in terms of bearing measurements. Then, the problem of the *agents motion coordination* is addressed: the goal is to steer the given formation along trajectories that do not affect the bearing measurements. In both cases, the quaternion-based $SE(3)$-bearing rigidity theory is exploited and the linear and

	rotation matrices	quaternions
nodes domain	$\boldsymbol{\chi}_i = (\mathbf{p}_i, \mathbf{R}_i) \in \mathbb{R}^3 \times SO(3)$	$\boldsymbol{\chi}_i^+ = (\mathbf{p}_i, \mathbf{q}_i) \in \mathbb{R}^3 \times \mathbb{S}^3$
nodes dynamics	$\dot{\mathbf{p}}_i = \mathbf{R}_i \mathbf{v}_i$ $\dot{\mathbf{R}}_i = \mathbf{R}_i [\boldsymbol{\omega}_i]_\times$	$\dot{\mathbf{p}}_i = \mathbf{R}(\mathbf{q})\mathbf{v}_i$ $\dot{\mathbf{q}}_i = \frac{1}{2}\mathbf{V}(\mathbf{q}_i)\boldsymbol{\omega}_i^+$
bearing measurements	$\mathbf{b}_{ij} = \mathbf{R}_i^\top \bar{\mathbf{p}}_{ij} \in \mathbb{S}^2$	$\mathbf{b}_{ij}^+ = \mathbf{q}_i^{-1} \odot \bar{\mathbf{p}}_{ij}^+ \odot \mathbf{q}_i = \begin{bmatrix} 0 \\ \mathbf{R}^\top(\mathbf{q}_i)\bar{\mathbf{p}}_{ij} \end{bmatrix} \in \mathbb{S}^3$
rigidity function	$\mathbf{b}_\mathcal{G} : SE(3)^n \to \mathbb{S}^{2m}$ $\boldsymbol{\chi} \to \mathbf{b}_\mathcal{G}(\boldsymbol{\chi}) = \begin{bmatrix} \mathbf{b}_1^\top \dots \mathbf{b}_m^\top \end{bmatrix}^\top$	$\mathbf{b}_\mathcal{G}^+ : SE(3)^n \to \mathbb{S}^{3m}$ $\boldsymbol{\chi}^+ \to \mathbf{b}_\mathcal{G}^+(\boldsymbol{\chi}^+) = \begin{bmatrix} (\mathbf{b}_1^+)^\top \dots (\mathbf{b}_m^+)^\top \end{bmatrix}^\top$
rigidity matrix	$\mathbf{B}_\mathcal{G}(\boldsymbol{\chi}) \in \mathbb{R}^{3m \times 6n}$ $[\mathbf{B}_\mathcal{G}(\boldsymbol{\chi})]_{k,i} = -\frac{P(\mathbf{b}_{ij})}{\\|\mathbf{p}_j - \mathbf{p}_i\\|_2} \in \mathbb{R}^{3\times 3}$ $[\mathbf{B}_\mathcal{G}(\boldsymbol{\chi})]_{k,j} = \frac{P(\mathbf{b}_{ij})}{\\|\mathbf{p}_j - \mathbf{p}_i\\|_2}\mathbf{R}_i^\top \mathbf{R}_j \in \mathbb{R}^{3\times 3}$ $[\mathbf{B}_\mathcal{G}(\boldsymbol{\chi})]_{k,3n+i} = [\mathbf{b}_{ij}]_\times \in \mathbb{R}^{3\times 3}$	$\mathbf{B}_\mathcal{G}^+(\boldsymbol{\chi}^+) \in \mathbb{R}^{4m \times 7n}$ $\left[\mathbf{B}_\mathcal{G}^+(\boldsymbol{\chi}^+)\right]_{k,i} = -\begin{bmatrix} 0 \\ \frac{P(\mathbf{b}_{ij})}{\\|\mathbf{p}_j - \mathbf{p}_i\\|_2} \end{bmatrix} \in \mathbb{R}^{4\times 3}$ $\left[\mathbf{B}_\mathcal{G}^+(\boldsymbol{\chi}^+)\right]_{k,j} = \begin{bmatrix} 0 \\ \frac{P(\mathbf{b}_{ij})}{\\|\mathbf{p}_j - \mathbf{p}_i\\|_2} \end{bmatrix} \mathbf{R}(\mathbf{q}_i)^\top \mathbf{R}(\mathbf{q}_j) \in \mathbb{R}^{4\times 3}$ $\left[\mathbf{B}_\mathcal{G}^+(\boldsymbol{\chi}^+)\right]_{k,3n+i} = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & [\mathbf{b}_{ij}]_\times \end{bmatrix} \in \mathbb{R}^{4\times 4}$

Table 7.1. $SE(3)$ bearing rigidity theory - rotation matrix based VS quaternion based formulation.

angular velocities (expressed in body frame) of each agent are assumed as control inputs, without considering any bounds.

7.4.1 Formation Stabilization

Given a $SE(3)$-bearing rigid framework $(\mathcal{G}, \boldsymbol{\chi}_0^+)$, the intent is to design a decentralized controller able to stabilize the formation to a desired configuration $\boldsymbol{\chi}_d^+ \in \mathbb{R}^{3n} \times \mathbb{S}^{3n}$ such that the resulting framework $(\mathcal{G}, \boldsymbol{\chi}_d^+)$ is BR and $\mathbf{b}_\mathcal{G}^+(\boldsymbol{\chi}_d^+)$ is the corresponding desired value for the $SE(3)$-bearing rigidity function.

Ensuring the fulfillment of the stabilization control task means guaranteeing the convergence of the bearing measurements to the desired ones stacked in $\mathbf{b}_\mathcal{G}^+(\boldsymbol{\chi}_d^+) \in \mathbb{S}^{3m}$. To this end, it is useful to take into account the $SE(3)$-bearing function dynamics, namely

$$\dot{\mathbf{b}}_\mathcal{G}^+(\boldsymbol{\chi}^+) = \mathbf{B}_\mathcal{G}^+(\boldsymbol{\chi}) \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega}^+ \end{bmatrix}, \tag{7.34}$$

where $\begin{bmatrix} \mathbf{v}^\top & (\boldsymbol{\omega}^+)^\top \end{bmatrix}^\top \in \mathbb{R}^{7n}$ corresponds to the formation commands input vector.

Being inspired by Schiano et al. (2016), a suitable control law that allows to stabilize the given formation to one BE to $(\mathcal{G}, \boldsymbol{\chi}_d^+)$ is given by

$$\begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega}^+ \end{bmatrix} = k_c (\bar{\mathbf{B}}_\mathcal{G}^+(\boldsymbol{\chi}^+))^\top \mathbf{b}_\mathcal{G}^+(\boldsymbol{\chi}_d^+) \tag{7.35}$$

where $\bar{\mathbf{B}}_\mathcal{G}^+(\boldsymbol{\chi}^+) \in \mathbb{R}^{4m \times 7n}$ is the *scale-free version* of the $SE(3)$-bearing rigidity matrix

$$\bar{\mathbf{B}}_\mathcal{G}^+(\boldsymbol{\chi}^+) = \left[\text{diag}\left(\left\{\mathbf{R}_i^\top P(\bar{\mathbf{p}}_{ij})\right\}\right) (\mathbf{E} \otimes \mathbf{I}_3)^\top \text{diag}\left(\{\mathbf{R}_i\}\right) \quad \text{diag}(\{[\mathbf{b}_{ij}]_\times\})(\mathbf{E}_\text{o} \otimes \mathbf{I}_3)^\top \right]. \tag{7.36}$$

Roughly speaking, this is obtained by multiplying the elements $[\mathbf{B}_{\mathcal{G}}^{+}(\boldsymbol{\chi})]_{k,i}$ and $[\mathbf{B}_{\mathcal{G}}^{+}(\boldsymbol{\chi})]_{k,j}$ (in (7.29) and (7.30), respectively) by the distance $\|\mathbf{p}_j - \mathbf{p}_i\|_2$, $\forall\ i, j \in \{1 \ldots n\}$ and $k \in \{1 \ldots m\}$. This scale-free version of the rigidity matrix is considered in order to avoid problems related to the unknown scale factor of the formation.

Conveniently in the multi-agent context, the controller (7.35) can be implemented in a distributed way as follows

$$\mathbf{v}_i = -k_c \sum_{(i,j)\in\mathcal{E}} \begin{bmatrix}\mathbf{0} & \mathbf{P}(\mathbf{b}_{ij})\end{bmatrix} (\mathbf{b}_{ij}^{+})_d + k_c \sum_{(j,i)\in\mathcal{E}} \mathbf{R}(\mathbf{q}_i^{-1} \odot \mathbf{q}_j) \begin{bmatrix}\mathbf{0} & \mathbf{P}(\mathbf{b}_{ji})\end{bmatrix} (\mathbf{b}_{ji}^{+})_d \tag{7.37}$$

$$\boldsymbol{\omega}_i^{+} = k_c \sum_{(i,j)\in\mathcal{E}} \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & [\mathbf{b}_{ij}]_{\times}^{\top} \end{bmatrix} (\mathbf{b}_{ij}^{+})_d \tag{7.38}$$

where $k_c \in \mathbb{R}^{+}$ is a tunable gain that influences the convergence rate of the control law.

Note that for each i-th agent, $\mathbf{b}_{ij}$ and $(\mathbf{b}_{ij}^{+})_d$ are locally available information, while $\mathbf{b}_{ji}, (\mathbf{b}_{ji}^{+})_d, \mathbf{p}_j, \mathbf{q}_j$ are quantities related to the neighboring j-th agent thus locally available to i-th agent up to 1-hop communication. On the other hand, $(\mathbf{p}_i, \mathbf{q}_i)$, namely the i-th agent pose, has to be estimated using a $SE(3)$ localization algorithm[4].

7.4.2 Coordinated Infinitesimal Bearing Rigid Motions

Besides the formation stabilization, the $SE(3)$-bearing rigidity results to be a suitable framework to handle a second control task that can be achieved independently from the previous one. This deals with the distributed government of the formation coordinated motions. In detail, the control purpose is to steer in a decentralized way the agents group along motion directions that do not affect the bearing relative measurements. Trivially, these coincide with the directions spanned by the null space of the rigidity matrix associated to the given IBR formation $(\mathcal{G}, \boldsymbol{\chi}^{+})$. Analyzing $\ker(\mathbf{B}_{\mathcal{G}}^{+}(\boldsymbol{\chi}^{+}))$ (equal to $\ker(\mathbf{B}_{\mathcal{K}}^{+}(\boldsymbol{\chi}^{+}))$ because $(\mathcal{G}, \boldsymbol{\chi}^{+})$ is assumed to be IBR), it is possible to characterize the formation bearing-preserving motions since they are identified by

$$\mathbf{n}_1' = \begin{bmatrix} \mathrm{diag}(\{\mathbf{R}(\mathbf{q}_i)^{\top}\})(\mathbf{1}_n \otimes \mathbf{I}_3) \\ \mathbf{0} \end{bmatrix} \qquad \mathbf{n}_2' = \begin{bmatrix} \mathrm{diag}(\{\mathbf{R}(\mathbf{q}_i)^{\top}\})\widetilde{\mathbf{p}} \\ \mathbf{0} \end{bmatrix}$$

$$\mathbf{n}_3' = \begin{bmatrix} \mathrm{diag}(\{\mathbf{R}(\mathbf{q}_i)^{\top}\}) \begin{bmatrix}\mathbf{E}_1\widetilde{\mathbf{p}} & \mathbf{E}_2\widetilde{\mathbf{p}} & \mathbf{E}_3\widetilde{\mathbf{p}}\end{bmatrix} \\ \mathrm{diag}(\{\mathbf{V}(\mathbf{q}_i^{-1})\mathbf{W}(\mathbf{q}_i)\})(\mathbf{1}_n \otimes [\mathbf{0}^{\top}\mathbf{I}_3]^{\top}) \end{bmatrix} \tag{7.39}$$

where the quantities $\mathbf{E}_i$ with $i \in \{1, 2, 3\}$ and $\widetilde{\mathbf{p}}$ have been defined in Corollary 7.2.10.

To realize the bearing-preserving motions (7.39) in a coordinated manner the following world frame velocity has to be implemented by the formation

$$\mathbf{s} = \mathbf{n}_1\boldsymbol{\nu} + \mathbf{n}_2\varrho + \mathbf{n}_3\boldsymbol{\mu}. \tag{7.40}$$

[4] A $SE(3)$-bearing rigidity based localization algorithm is not presented in this manuscript because this is still an ongoing research project.

This imposes to the framework a common linear velocity $\boldsymbol{\nu} \in \mathbb{R}^3$, an expansion rate $\varrho \in \mathbb{R}$ about the formation barycentre, and a coordinated rotation with angular speed $\boldsymbol{\mu} \in \mathbb{R}^3$ about a generic axis passing through the formation barycentre.

Adopting the distributed paradigm, the decentralized control scheme for the formation is such that each i-th agent has to realize the following linear and angular velocity

$$\mathbf{v}_i = \mathbf{R}^\top(\mathbf{q}_i)(\boldsymbol{\nu} + \varrho(\mathbf{p}_i - \bar{\mathbf{p}}) + \boldsymbol{\mu}[\mathbf{p}_i - \mathbf{p}^*]_\times) \tag{7.41}$$

$$\boldsymbol{\omega_i}^+ = \mathbf{R}^\top(\mathbf{q}_i)\boldsymbol{\mu} \tag{7.42}$$

where the difference $\mathbf{p}_i - \mathbf{p}^*$ accounts for the distance of the robot from the system CoM.

Remark 7.4.1. The control law (7.40) can be implemented jointly with (7.35) without affecting the achievement of the two tasks. Nevertheless, it is a common practice to achieve the formation stabilization before steering it along the desired trajectories since this ensures small tracking errors during the transient phase.

7.5 Numerical Results

To validate the controllers (7.35)-(7.40), numerical simulations have been carried out taking into account a formation of $n = 8$ fully-actuated agents. All the runs start from the same initial conditions $\boldsymbol{\chi}^+(t_0) = \boldsymbol{\chi}_0^+ \in SE(3)^8$ so that the robots positions coincide with the vertexes of a cube and they are all oriented in the same manner. Specifically, it is assumed that $\mathbf{q}_i(t_0) = \mathbf{q}_I$ for $i = 1 \dots 8$. Different topologies are instead evaluated concerning the underlying communication and sensing graph.

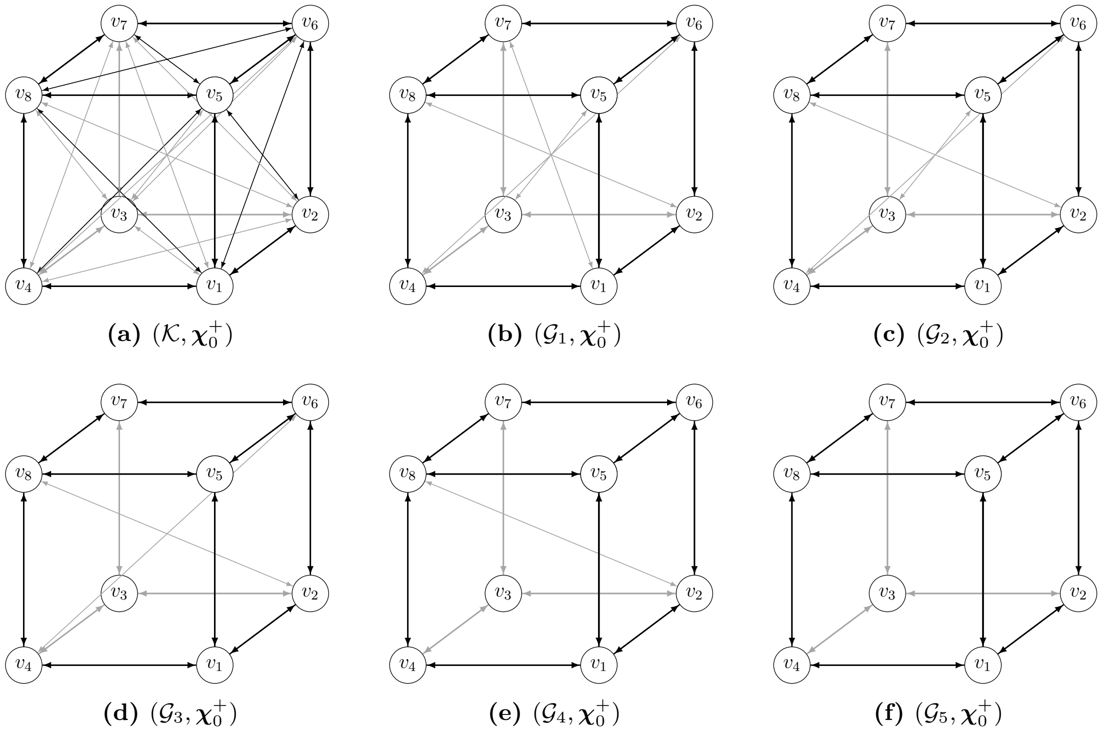
7.5.1 Test 1: Formation Stabilization

To prove the effectiveness of the proposed solution (7.35), the control error is introduced and defined as the norm of the difference between the desired bearing measurements stacked in $\mathbf{b}_\mathcal{G}^+(\boldsymbol{\chi}_d^+)$ and the current ones, normalized over the number of edges (available measurements) in the graph. Formally, it is

$$e_s(t) = \frac{1}{m}\|\mathbf{b}_\mathcal{G}^+(\boldsymbol{\chi}^+(t)) - \mathbf{b}_\mathcal{G}^+(\boldsymbol{\chi}_d^+)\|_2. \tag{7.43}$$

In addition, the level of rigidity of the considered frameworks is taken into account for the purpose of investigating its relation with the convergence properties of the controller. In Trinh, Park, Sun, Anderson, Pham, and Ahn (2016), two rigidity indexes are proposed resting upon the fact that the infinitesimal rigidity property of a framework is connected to the rank of the bearing rigidity matrix. Following the same reasoning, given a IBR framework $(\mathcal{G}, \boldsymbol{\chi}^+)$, let first introduce the matrix

$$\mathbf{B}_\mathcal{G}^{sym}(\boldsymbol{\chi}^+) = (\mathbf{B}_\mathcal{G}^+(\boldsymbol{\chi}^+))^\top \mathbf{B}_\mathcal{G}^+(\boldsymbol{\chi}^+) \in \mathbb{R}^{7n \times 7n}. \tag{7.44}$$



(a) $(\mathcal{K}, \boldsymbol{\chi}_0^+)$ **(b)** $(\mathcal{G}_1, \boldsymbol{\chi}_0^+)$ **(c)** $(\mathcal{G}_2, \boldsymbol{\chi}_0^+)$

(d) $(\mathcal{G}_3, \boldsymbol{\chi}_0^+)$ **(e)** $(\mathcal{G}_4, \boldsymbol{\chi}_0^+)$ **(f)** $(\mathcal{G}_5, \boldsymbol{\chi}_0^+)$

Figure 7.5. $SE(3)$ frameworks considered in the numerical simulations: the poses of the agents are always the same, while the underlying graph changes.

This, termed *symmetric rigidity matrix* in Zelazo et al. (2012, 2015a) and *vertex rigidity Gramian* in Shames and Summers (2015), has trivially the same rank of $\mathbf{B}_{\mathcal{G}}^{+}(\boldsymbol{\chi}^+)$, and, as a consequence, it has also $n+7$ zero eigenvalues. Thus, to evaluate the rigidity level of $(\mathcal{G}, \boldsymbol{\chi}^+)$, the following two indexes can be considered.

- *Worst-Case index* (WC-index) is defined as the ratio between the smallest non-zero eigenvalue of $\mathbf{B}_{\mathcal{G}}^{sym}(\boldsymbol{\chi}^+)$, namely λ_{n+8}, and its trace, i.e., the sum of all its eigenvalues. Formally, this is

$$i_{WC} = \frac{\lambda_{n+8}}{\operatorname{tr}(\mathbf{B}_{\mathcal{G}}^{sym}(\boldsymbol{\chi}^+))}. \tag{7.45}$$

- *IMBalance index* (IMB-index) is computed as the ratio of the $(n+8)$-th and the largest eigenvalue of $\mathbf{B}_{\mathcal{G}}^{sym}(\boldsymbol{\chi}^+)$, namely

$$i_{IMB} = \frac{\lambda_{n+8}}{\lambda_{7n}}. \tag{7.46}$$

The efficiency of the stabilization control solution (7.35) is confirmed by the results reported in Figure 7.6, where the trend of the control error is reported for five different $SE(3)$ frameworks. The underlying graph indeed varies according to Figures 7.5a-7.5e, passing from the directed complete graph $\mathcal{K}$ to the topology $\mathcal{G}_4$ where the cut of edges (2,8) and (8,2) (Figure 7.5f) causes the loss of the infinitesimal bearing rigidity property.

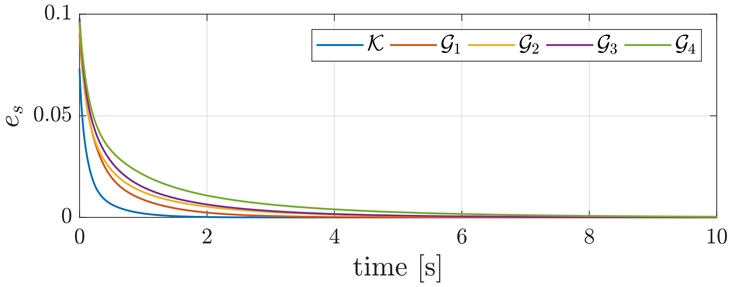


Figure 7.6. Test 1: control error for the stabilization of a given rigid formation.

	$(\mathcal{K},\boldsymbol{\chi}_0^+)$	$(\mathcal{G}_1,\boldsymbol{\chi}_0^+)$	$(\mathcal{G}_2,\boldsymbol{\chi}_0^+)$	$(\mathcal{G}_3,\boldsymbol{\chi}_0^+)$	$(\mathcal{G}_4,\boldsymbol{\chi}_0^+)$
i_{WC}	0.0043	0.0024	0.0019	0.0013	0.0007
i_{IMB}	0.0864	0.0467	0.0348	0.0227	0.0124

Table 7.2. Test 1: WC-index and IMB-index for the considered frameworks at $t = t_0$.

The control goal is to stabilize the given framework in a desired IBR configuration which is not reachable implementing a sequence of infinitesimal motions, namely a structure where the agents are placed in the vertexes of another parallelepiped while keeping constant the robots attitude. It is assumed that the bearing measurements are perturbed by additive Gaussian noise having unit variance, while the parameter k_c is set to 1.

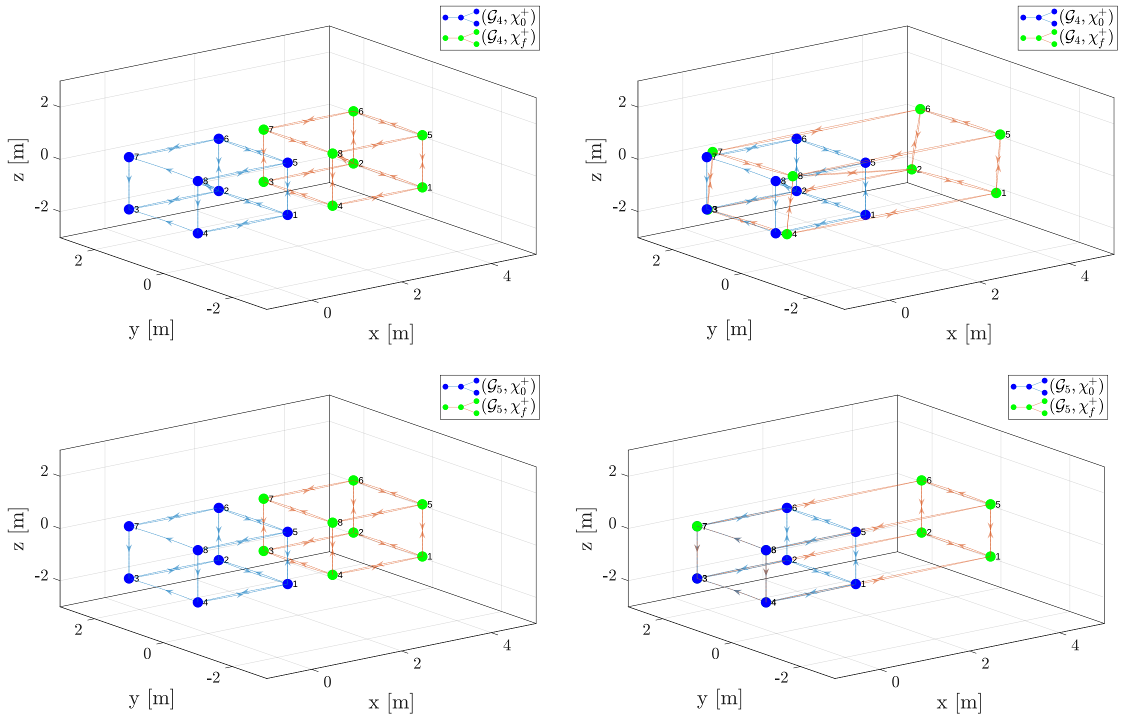
Figure 7.6 shows that in all the cases the formation reaches a configuration that is BE to the desired one, namely the control error e_s converges towards zero. It is interesting to note that the rigidity level of the framework influences the convergence rate of the error. Specifically, one can observe that the convergence rate increases in a proportional manner with respect to the rigidity indexes computed at t_0 and reported in Table 7.2. This observation is in-line with the intuition that the efficiency of the controller generally depends on the amount of the available information.

7.5.2 Test 2: Coordinated Motions

To study the performance of the controller (7.40), the frameworks associated to the topologies $\mathcal{G}_4$ in Figure 7.5e and $\mathcal{G}_5$ in Figure 7.5f are considered. Note that $(\mathcal{G}_4,\boldsymbol{\chi}_0^+)$ results to be infinitesimal bearing rigid, unlike $(\mathcal{G}_5,\boldsymbol{\chi}_0^+)$.

First, it can be observed that the bearing measurements are always preserved when the formations are steered along the directions identified by the null space of the bearing rigidity matrix $\mathbf{B}^+_{\mathcal{G}_4}(\boldsymbol{\chi}_0^+)$. Specifically, Figures 7.7a illustrate the case of a translation along the x-axis of all the agents. It occurs that $\mathbf{b}_{\mathcal{G}_4}(\boldsymbol{\chi}_0^+) = \mathbf{b}_{\mathcal{G}_4}(\boldsymbol{\chi}_f^+)$ and $\mathbf{b}_{\mathcal{G}_5}(\boldsymbol{\chi}_0^+) = \mathbf{b}_{\mathcal{G}_5}(\boldsymbol{\chi}_f^+)$, where $\boldsymbol{\chi}^+(t_f) = \boldsymbol{\chi}_f^+$ denotes the final configuration of the frameworks: the relative bearing measurements are preserved for both the evaluated cases.

Conversely, translating only some robots keeping fixed the others, the structure of the framework related to $\mathcal{G}_4$ distorts, while the bearing measurements are preserved when the topology $\mathcal{G}_5$ is considered. Figures 7.7b depict these results which can be explained



(a) rigid motion

(b) non-rigid motion

Figure 7.7. Test 2: coordinated rigid (a) and non-rigid (b) motions for an infinitesimally rigid framework and for a roto-flexible framework.

observing that the realized motion does not belong to the null space of $\mathbf{B}^{+}_{\mathcal{G}_4}(\boldsymbol{\chi}^{+}_0)$ (equal to $\mathbf{B}^{+}_{\mathcal{K}}(\boldsymbol{\chi}^{+}_0)$), but it is in the kernel of $\mathbf{B}^{+}_{\mathcal{G}_5}(\boldsymbol{\chi}^{+}_0)$, thus guaranteeing the bearing measurements preservation for the BRF framework $(\mathcal{G}_5, \boldsymbol{\chi}^{+}_0)$.

7.6 Results Summary

This chapter introduces an extension of the rigidity theory to the case of frameworks embedded in $SE(3)$, which represent a suitable model for fully-actuated UAVs swarms. The main contribution is the characterization of the infinitesimal rigidity properties that are related to the rank of the $SE(3)$- bearing rigidity matrix. Moreover, it has been showed that the null space of that matrix describes the infinitesimal motions of a $SE(3)$ framework, which correspond to the rigid body translations and uniform scaling, in addition to 3D coordinated rotations, when the framework is infinitesimally bearing rigid. All these concepts are derived assuming to represent the agents attitude using the rotation matrices formalism.

Then, these are redefined by using the quaternion representation of the vehicles orientation. This choice is motivated by the consequent simplification in the design of a formation control whose aim is twofold. Given a rigid multi-agent system, the purpose is to steer it into a desired configuration and then to drive the formation along certain directions that ensure the bearing measurements maintenance.

The effectiveness of the distributed proposed controllers has been validated through a simulative campaign taking into account formations related to various underlying topologies. Interestingly, the convergence toward a desired configuration emerges to be dependent on the rigidity level of the framework.

8

Conclusions

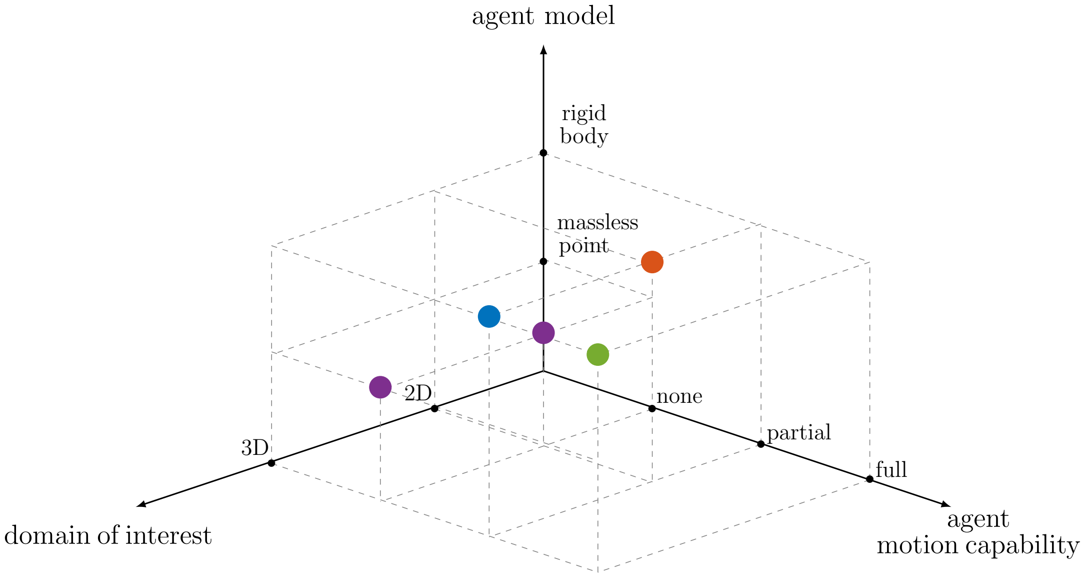


Figure 8.1. Graphical summary of the multi-agent systems considered in this thesis: wireless sensor networks (purple), planar (orange) and three-dimensional (blue) camera systems, aerial platform formations (green).

8.1 Conclusions

This dissertation deals with estimation and control applications for multi-agent systems required to cooperatively and autonomously act in smart environments. Three main parts can be distinguished according to the type of considered technology: (industrial) wireless sensor networks, multi-camera systems and aerial platform formations.

Figure 8.1 aims at arranging the evaluated multi-device architectures in the help scheme introduced in Chapter 1 that, as regards for each agent in the group, accounts for the used model, the motion capability and the domain of interest.

8.1.1 First Part

The first part, coinciding with Chapter 2, deals with wireless sensor networks (WSNs), especially used in the automated industrial scenario. These multi-agent systems are composed by a large number of tiny smart sensors modeled as massless points, provided with communication capabilities over wireless links but having fixed position in the (two or three-dimensional) environment of interest.

In Chapter 2, clustering issue is faced for a given industrial wireless sensor network (IWSN). This task has been formally stated as the determination of the unique partition that simultaneously abides by three different criteria. First, for each pair of nodes, in a cluster, there must exist at least one (communication) path connecting them composed exclusively of elements included in the same cluster (*connectivity criterion*). Second, for each node in a cluster, there must exist at least another one in the same cluster, such that the two nodes measurements are similar according to a defined metric (*measurements* *similarity criterion*). Finally, the network must be partitioned into the minimal number of non-overlapping clusters (*maximality criterion*).

The most important and original contribution has been the design of algorithms that solve the aforementioned clustering task. Specifically, the partitioning problem has been solved, in the first instance, through a centralized policy and then, the distributed paradigm has been considered to provide an algorithm capable of partitioning the network only by local exploitation of measurements. In both cases, no cluster heads assignment has been considered, and no assumptions have been made on the structure of the emerging clusters (other than those implied by the partitioning criteria), which are uniquely determined by the proposed algorithms. The convergence of the distributed solution to the centralized one has been showed through a numeric example and the efficiency of the proposed approach has been confirmed by comparing it with other clustering techniques, i.e., a classical k-means strategy (Hastie, Tibshirani, and Friedman (2009)) and the most recent DMGA algorithm (Chen et al. (2015)). Finally, the applicability of the proposed procedure to real industrial use cases has been demonstrated by experimental and simulated scenarios. As a first example, the decentralized algorithm (DCA) has been employed to perform anomaly detection in a simulated industrial production process. Subsequently, the performances of a fault detection algorithm based on this procedure have been evaluated on an environmental sensing dataset and compared with those offered by the state-of-the-art GAD algorithm, that serves the same purpose.

8.1.2 Second Part

The second part of this thesis, namely Chapter 3 and Chapter 4, is focused on the multi-camera systems (visual sensor networks, VSNs) for surveillance applications. Two possibilities have been distinguished: (i) camera networks modeled as a set of rigid bodies in the 3D space whose orientation could be varied over time exploiting three DoFs, and (ii) planar camera networks acting in the 2D space so that their orientation is univocally determined by their pan angle.

Chapter 3 is devoted to the solution of the attitude estimation problem for both three-dimensional and planar multi-camera systems dealing with the reconstruction of the orientation of each device in the group with respect to a fixed global reference frame. For the cameras acting in the 3D space, this translates into the determination of n rotation matrices belonging to the Special Orthogonal group $SO(3)$, while in the 2D case n pan angles have to be recovered, being n the number of devices composing the system. To this aim the TV-solution has been considered (Tron and Vidal (2009, 2014)) which rests upon the iterative minimization of a suitable cost function defined over the proper manifold.

Regarding the 3D systems, the main contribution has been the definition and comparison of several initialization methods for the aforementioned optimization procedure, which ensure the convergence to an optimal solution starting from the noisy measurements of the relative poses. Specifically, various strategies based on different ways to explore the network have been proposed with the aim of correctly initializing the iterative minimization by exploiting the available information. Most interestingly, the goodness of the achieved results on synthetic and real scenarios suggests that the proposed methods can also be exploited by them selves as a centralized solution for the rotation estimation problem. Actually, a second contribution of interest has regarded the implementation of the distributed procedure on a real-world scenario, thus pointing out the effectiveness of the proposed algorithms with a realistic distribution of the measurement noise.

Concerning the planar case, the restriction of the orientation estimation problem to a planar domain is so that the determination of a rotation matrix $\mathbf{R} \in SO(2)$ corresponds to the determination of a single rotation angle, thus requiring for the solution of an angular synchronization problem. The main contribution derived from this simplification is the emergence of an elegant connection between the minimization of a convex functional that yields the optimal rotation estimates and the theory of certain stochastic matrices, providing an insight on the role of the topological links that define the neighborhood of a camera node. In doing this, an iterative estimation procedure through two different update rules has been proposed. The convergence properties of these strategies rely on the eigenvalue analysis of a particular class of stochastic matrices which govern the iterative scheme and derive from the network topology. The numerical simulation results support the theoretical considerations about the convergence performance highlighting the role of bipartite graphs.

Chapter 4 handles the problem of perimeter patrolling within the surveillance context though a planar camera system (one-dimensional problem). Following the steps paved by Carli et al. (2011) and similar works, the issue has been recast as the determination of an optimal partition for the perimeter to monitor, by minimizing the time interval between two successive inspections of the same point, i.e., the lag time. At the same time, though, other fundamental aspects are considered, that constitute the main contributions. First, the patrolling problem has been addressed by modeling the perimeter as a closed curve. In this sense, the novelty aspect of the proposed approach rests upon the solution of a perimeter partition rather than a segment partition, differently from the previous literature. Then, the optimality criterion has been enriched introducing an additional penalty function related to the quality of vision of the cameras on the perimeter, i.e., to the angle of visual impact. The resulting regularized optimization problem has been analytically solved. To conclude, a distributed algorithm has been proposed that exploits local communication among the devices and iteratively converges to the optimal solution. Such an algorithm has been tested in a simulative campaign involving different setups and levels of noise, showing excellent performance when compared with the current state-of-the-art represented by the s-PA solution (Alberton et al. (2012)).

8.1.3 Third Part

The third part of this manuscript is composed by Chapter 5, Chapter 6 and Chapter 7 wherein aerial platforms (UAVs) are considered. Such kind of vehicles are characterized by full motion capability: their kinematics is given by the physics of rigid bodies which are able to translate and rotate in the 3D space according to six DoFs. Note that a multi-rotor UAV can be part of a multi-agent system when it belongs to a swarm, but it can be also considered as a multi-agent itself where the rotors play the role of the agents. The first interpretation is adopted in this thesis, although initially (Chapter 5-6) the attention has been focused on the design and control of a single platform.

Introducing an appropriate dynamic model for the class of the Generically-Tilted Multi-rotors, Chapter 5 has first investigated the coupling between the control force and the control moment that emerges from the intrinsic cascaded dependency of the UAVs translational dynamics on the rotational one. Some necessary algebraic conditions have been derived on the control input space that imply the possibility to independently act on the vehicle position and attitude. More in detail, it has been distinguished among the existence of a preferential direction, the existence of a preferential plane and the full actuation condition: while in the first two cases the control force can be assigned independently of the control moment only along certain directions, when the platform is fully actuated then it has six controllable DoFs. To validate the given statements, the fulfillment of these properties has been analyzed for platforms known in the literature, categorizing them through the provided taxonomy. As a second contribution, the concept of rotor-failure robustness has been formalized based on the possibility for a multi-rotor to realize static hovering even in case a propeller fails and stops spinning. Specifically, static hovering has been defined as an equilibrium flight condition wherein the platform stays in a constant spot with zero linear and angular velocity. According to the given robustness definition, an extensive discussion on the vulnerability properties of the hexarotor platforms has been carried out. To do this, the vehicle structure has been parametrized by three angles that determine the positions (with respect to the platform center of mass) and the spinning axes direction of the six propellers in order to span the most known classes of six-rotor UAVs. The study of the role of these angles has showed that both the tilted star-shaped hexarotor and Y-shaped hexarotor can still hover statically after a rotor-loss.

In Chapter 6 two static hovering control solutions have been presented whose aim is to steer and keep a given generically-tilted multi-rotor platform in a constant reference position ensuring a constant attitude at the steady-state. Both controllers require the existence of a preferential direction in the force space along which the control force can be freely assigned with respect to the control moment. As regard for the existing state-of-the-art controllers, it is assumed that this may be arbitrarily oriented. The first proposed controller exploits a sort of dynamic feedback linearization around such a preferential direction and uses the quaternion formalism to represent the vehicle attitude. The asymptotic convergence of the error dynamics has been proved using a hierarchical Lyapunov-based approach over nested sets and has been confirmed by simulation results on a hexarotor with tilted propellers. The second control solution instead constitutes a generalization of the cascaded control for the case in which the control force and moment are not fully decoupled, as it usually happens when a failure occurs. The effectiveness of this control strategy has been confirmed by the simulations and real experiments in the failure scenario. These tests have also validated the previous outcomes about the robustness properties of the hexarotors.

Finally, in Chapter 7, two control objectives have been addressed in the $SE(3)$ space considering formations of fully-actuated aerial platforms: first, the stabilization of the multi-agent system towards a desired configuration, and then the coordinated motion along directions preserving the inter-agents bearing measurements, i.e., those described by the null space of the bearing rigidity matrix associated to the framework. To this end, as first step, the study of the bearing rigidity theory has been extended to frameworks embedded in the $SE(3)$ space. As for the $SE(2)$ case, the principal features of such theory concern the bearing measurements that are referred to the local frame of each agent, corroborating the use of a directed graph to represent agents interactions. Nevertheless, the extension from the two-dimensional Special Euclidean space to the three-dimensional one is not trivial. The main challenge deals with the attitude representation of the agents: while in $SE(2)$ a single angle is sufficient to determine the orientation of a robot on the plane, in 3D space three DoFs have to be managed involving different and more complex types of manifolds. In this sense, even though the trivial infinitesimal motions are still translations, uniform scaling and coordinated rotations, they have been redefined in the new higher-dimensional context. In addition, it has been proved that these are related to the null space of the $SE(3)$-bearing rigidity matrix of each generic framework and, similarly to Zelazo et al. (2014), that a framework is infinitesimally rigid in $SE(3)$ if and only if these motions constitute the unique bearing-preserving deformations, or equivalently, the size of the null space of the $SE(3)$-bearing rigidity matrix is equal to seven. Then a new formulation for the bearing rigidity theory has been proposed by representing the agents attitude with the unit quaternions in place of the rotation matrices. Such a formulation facilitates the design of a distributed control law to stabilize a given rigid formation of fully-actuated robots to a configuration having the desired bearing measurements. Moreover, a second decentralized solution has been proposed to realize coordinated formation motions along the bearing-preserving directions. The effectiveness of both the rigidity-based control strategies has been proved by the results of a simulative campaign. Note that the addressed control tasks could be simultaneously solved being orthogonal among them.

8.1.4 Discussion

The reader can note how the manuscript evolves from the management of fixed sensor networks to the control of fully-actuated aerial platform formations, moving along the axis of the agent motion capability in Figure 8.1. In this sense, the application complexity increases chapters by chapters having to deal with an increasing number of DoFs.

This observation is enforced by Table 8.1 that aims at resuming the manuscript contents, highlighting the complexity evolution in the marked column. The first column of the table reports the subdivision in parts of the dissertation. From the third to the sixth columns, the main characteristics of the devices composing the considered

	multi-agent system	agent properties				application	
		location description	variable quantities	acquired data	neighboring interactions		
1	WSN	$\mathbf{p}_i \in \mathbb{R}^d$ $d \in \{2,3\}$	none	$m_i \in \mathbb{R}$	bidirectional	clustering strategy Ch. 2	
2	2D VSN	$\mathbf{p}_i \in \mathbb{R}^2$ $\mathbf{R}_i \in SO(2)$ $(\gamma_i \in \mathbb{S}^1)$	$\mathbf{R}_i \in SO(2)$	2D images	bidirectional	attitude estimation Ch. 3	perimeter patrolling Ch. 4
	3D VSN	$\mathbf{p}_i \in \mathbb{R}^3$ $\mathbf{R}_i \in SO(3)$	$\mathbf{R}_i \in SO(3)$	2D images	bidirectional	attitude estimation Ch. 3	
3	UAVs swarm	$\mathbf{p}_i \in \mathbb{R}^3$ $\mathbf{R}_i \in SO(3)$ $(\mathbf{q}_i \in \mathbb{S}^3)$	$\mathbf{p}_i \in \mathbb{R}^3$ $\mathbf{R}_i \in SO(3)$ $(\mathbf{q}_i \in \mathbb{S}^3)$	3D bearings	unidirectional	static hovering Ch. 6	formation control Ch. 7

Table 8.1. Summary of the manuscript contents.

multi-agent system are summarized. In accordance with Section 1.2.1, the reported features are the mathematical description of the agent location in the domain of interest, the physical parameters that it can modify over time according to its motion capability, the typology of the sensed data and the interactions modality with the others agents in the group. Specifically, when devices information exchange is bidirectional the system underlying graph is undirected, while it is directed when the communication/sensing interaction is unidirectional. Note that there is an evolution from the one-dimensional measurements gathered in wireless sensors networks to the three-dimensional bearing vectors recovered by the aerial platforms exploiting the on-board sensors. However, the challenging aspect within the context of the multi-agent systems control regards the growth of the numbers of the controllable variables. Although this fact allows to potentially realize more sophisticated tasks in the optic of the creation of efficient smart environments, the dynamics and the kinematics of the agents become more complex implying the need of more refined control techniques.

8.2 Future works

Several future developments can be suggested according to the considered multi-agent system and its specific application.

Regarding the clustering strategies for WSNs, future research directions might be focused on the real-time implementation of the proposed algorithms in a testbed wireless sensor network to investigate the practical challenges that could arise from their adoption. Moreover, future developments include the theoretical and simulative comparison of the proposed methods in more specific scenarios, as well as the deployment of such procedures in actual industrial applications.

With respect to the attitude estimation for the camera systems, it will be interesting to find some complexity bounds when modelling the network via certain classes of network topology that appear interesting for distributed surveillance and scene reconstruction applications. Furthermore, some additional study might be desirable for investigating the choice of the weights used in the computation of the rotation average, as well as the role of noise. Finally, a deeper analysis on real datasets can be conducted in order to strongly validate the described methods.

Concerning the perimeter patrolling, future works might involve the study of more specific topology issues in terms on both visibility (e.g., the management of occlusions in the field of view) and communication graph (e.g., asynchronous or lossy communication protocols). Interesting additional developments could include the integration of phase synchronization techniques for the cameras in order to perform some more complex tasks but complementary to patrolling, as the detection and tracking of intruders. Finally, the most challenging and appealing future work consists in the extension of the developed solution for the area patrolling task (two-dimensional problem).

For what regards the aerial platforms, in future, it will be interesting to apply the developed theory to assess the robustness of other classes of vehicles with more than six rotors and/or whose angular parameters can change during flight. In this perspective, the challenge is to design a n-rotor platform that is (fully) robust to the loss of $(n-4)$ propellers, e.g., a fully 2-losses robust hexarotor, or a fully 3-losses robust eptarotor, or a fully 4-losses robust octorotor. In addition, the control goal might be extended to the tracking of a dynamic position trajectory, even in failure scenario.

In the light of this discussion, the future research directions result to be multiple, stressing the idea that smarter environments are realizable. On the other hand, the results achieved in this thesis confirm the fact that multi-agent systems are a strategic technology towards the development of smarter cities where the improvement of all the aspects of the inhabitants life is the final goal.

A

Supplementary Material

A.1 Proof of the identity $\dot{\boldsymbol{\omega}}_d = \boldsymbol{\omega}_{dd}$

The identity $\dot{\boldsymbol{\omega}}_d = \boldsymbol{\omega}_{dd}$ stated in Section 6.2 is justified by the following equivalences where it has been exploited also the relation

$$\begin{aligned}
[[\boldsymbol{\epsilon}_1]_\times \boldsymbol{\epsilon}_2]_\times &= [\boldsymbol{\epsilon}_1]_\times [\boldsymbol{\epsilon}_2]_\times - [\boldsymbol{\epsilon}_2]_\times [\boldsymbol{\epsilon}_1]_\times \\
&= \boldsymbol{\epsilon}_2 \boldsymbol{\epsilon}_1^\top - \boldsymbol{\epsilon}_1^\top \boldsymbol{\epsilon}_2 \mathbf{I}_3 - \boldsymbol{\epsilon}_1 \boldsymbol{\epsilon}_2^\top + \boldsymbol{\epsilon}_2^\top \boldsymbol{\epsilon}_1 \mathbf{I}_3 \\
&= \boldsymbol{\epsilon}_2 \boldsymbol{\epsilon}_1^\top - \boldsymbol{\epsilon}_1 \boldsymbol{\epsilon}_2^\top .
\end{aligned} \tag{A.1}$$

Because of (6.15), the derivative of $\boldsymbol{\omega}_d$ results from the sum of three components, namely

$$\dot{\boldsymbol{\omega}}_d = \dot{\boldsymbol{\omega}}_{d,1} + \dot{\boldsymbol{\omega}}_{d,2} + \dot{\boldsymbol{\omega}}_{d,3} \tag{A.2}$$

with

$$\dot{\boldsymbol{\omega}}_{d,1} = -\frac{1}{\xi f^2}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu}\dot{f} \tag{A.3}$$

$$\overset{(6.16)}{=} -\frac{1}{(\xi f)^2}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu}\mathbf{d}_*^\top \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu} = -\frac{\left(\mathbf{d}_*^\top \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu}\right)}{(\xi f)^2}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu} \tag{A.4}$$

$$\dot{\boldsymbol{\omega}}_{d,2} = \frac{1}{\xi f}[\mathbf{d}_*]_\times \dot{\mathbf{R}}^\top(\mathbf{q}_d)\boldsymbol{\nu} \tag{A.5}$$

$$\overset{(5.1)}{=} -\frac{1}{\xi f}[\mathbf{d}_*]_\times [\boldsymbol{\omega}_d]_\times \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu} \tag{A.6}$$

$$\overset{(6.15)}{=} -\frac{1}{(\xi f)^2}[\mathbf{d}_*]_\times \left[[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu}\right]_\times \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu} \tag{A.7}$$

$$\overset{(A.1)}{=} -\frac{1}{(\xi f)^2}[\mathbf{d}_*]_\times \left(\mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu}\mathbf{d}_*^\top - \mathbf{d}_*\boldsymbol{\nu}^\top \mathbf{R}(\mathbf{q}_d)\right)\mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu} \tag{A.8}$$

$$= -\frac{1}{(\xi f)^2}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu}\mathbf{d}_*^\top \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu} = -\frac{\left(\mathbf{d}_*^\top \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu}\right)}{(\xi f)^2}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu} \tag{A.9}$$

$$\dot{\boldsymbol{\omega}}_{d,3} = \frac{1}{\xi f}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\dot{\boldsymbol{\nu}} \tag{A.10}$$

$$\overset{(6.17)}{=} \frac{1}{\xi f}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\left(\frac{k_{pd}k_{pp}}{m}\dot{\mathbf{e}}_p + \left(\frac{k_{pd}^2}{m} - k_{pp}\right)\dot{\mathbf{e}}_v - \left(\frac{k_{pd}}{m} + k_\Delta\right)\dot{\mathbf{f}}_\Delta\right) \tag{A.11}$$

$$\begin{aligned}
&\overset{(6.29)}{=} \frac{1}{\xi f}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\left(\frac{k_{pd}k_{pp}}{m}\mathbf{e}_v - \left(\frac{k_{pd}}{m} + k_\Delta\right)\dot{\mathbf{f}}_\Delta + \right. \\
&\qquad \left. + \left(\frac{k_{pd}^2}{m^2} - \frac{k_{pp}}{m}\right)\left(-mg\mathbf{e}_3 + (\mathbf{R}(\mathbf{q}) - \mathbf{R}(\mathbf{q}_d))\mathbf{d}_*\xi f + \mathbf{f}_r + \mathbf{f}_\Delta\right)\right)
\end{aligned} \tag{A.12}$$

$$\begin{aligned}
&= \frac{1}{\xi f}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\left(\frac{k_{pd}k_{pp}}{m}\mathbf{e}_v - \left(\frac{k_{pd}}{m} + k_\Delta\right)\dot{\mathbf{f}}_\Delta + \right. \\
&\qquad \left. + \left(\frac{k_{pd}^2}{m^2} - \frac{k_{pp}}{m}\right)\left(-mg\mathbf{e}_3 + \mathbf{R}(\mathbf{q})\mathbf{d}_*\xi f + \mathbf{f}_r + \mathbf{f}_\Delta\right)\right)
\end{aligned} \tag{A.13}$$

$$
\begin{aligned}
&\overset{(6.11)}{=} \frac{1}{\xi f}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\left(\frac{k_{pd}k_{pp}}{m}\mathbf{e}_v - \left(\frac{k_{pd}}{m}+k_\Delta\right)\dot{\mathbf{f}}_\Delta + \right.\\
&\qquad\qquad \left. + \left(\frac{k_{pd}^2}{m^2}-\frac{k_{pp}}{m}\right)(\mathbf{R}(\mathbf{q})\mathbf{d}_*\xi f - k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v + \mathbf{f}_\Delta)\right)
\end{aligned} \tag{A.14}
$$

$$
\begin{aligned}
&\overset{(6.44)}{=} \frac{1}{\xi f}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\left(\frac{k_{pd}k_{pp}}{m}\mathbf{e}_v + \left(\frac{k_{pd}}{m}+k_\Delta\right)k_\Delta\mathbf{f}_\Delta + \right.\\
&\qquad\qquad \left. + \left(\frac{k_{pd}^2}{m^2}-\frac{k_{pp}}{m}\right)(\mathbf{R}(\mathbf{q})\mathbf{d}_*\xi f - k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v + \mathbf{f}_\Delta)\right)
\end{aligned} \tag{A.15}
$$

Let define $\kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta) \in \mathbb{R}$ such that

$$
\kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta) = -\frac{2}{\xi f}\left(\mathbf{d}_*^\top \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu}\right) \tag{A.16}
$$

$$
= -\frac{2}{\xi f}\mathbf{d}_*^\top \mathbf{R}^\top(\mathbf{q}_d)\left(\frac{k_{pd}k_{pp}}{m}\mathbf{e}_p + \left(\frac{k_{pd}^2}{m}-k_{pp}\right)\mathbf{e}_v - \left(\frac{k_{pd}}{m}+k_\Delta\right)\mathbf{f}_\Delta\right) \tag{A.17}
$$

Note that it follows that

$$
\dot{\boldsymbol{\omega}}_{d,1} + \dot{\boldsymbol{\omega}}_{d,2} = -\frac{2}{(\xi f)^2}\left(\mathbf{d}_*^\top \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu}\right)[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu} \tag{A.18}
$$

$$
= -\frac{1}{\xi f}\kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\boldsymbol{\nu} \tag{A.19}
$$

$$
\overset{(6.17)}{=} -\frac{1}{\xi f}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\Big(\kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\,\frac{k_{pd}k_{pp}}{m}\mathbf{e}_p + \tag{A.20}
$$

$$
+\kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\left(\frac{k_{pd}^2}{m}-k_{pp}\right)\mathbf{e}_v - \kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\left(\frac{k_{pd}}{m}+k_\Delta\right)\mathbf{f}_\Delta\Big) \tag{A.21}
$$

and then

$$
\dot{\boldsymbol{\omega}}_d = \dot{\boldsymbol{\omega}}_{d,1} + \dot{\boldsymbol{\omega}}_{d,2} + \dot{\boldsymbol{\omega}}_{d,3} \tag{A.22}
$$

$$
= \frac{1}{\xi f}[\mathbf{d}_*]_\times \mathbf{R}^\top(\mathbf{q}_d)\left(\left(\frac{k_{pd}^2}{m^2}-\frac{k_{pp}}{m}\right)\mathbf{R}(\mathbf{q})\mathbf{d}_*\xi f + \right. \tag{A.23}
$$

$$
-\left(\frac{k_{pd}^2 k_{pp}}{m^2}+\frac{k_{pp}^2}{m}+\kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\,\frac{k_{pd}k_{pp}}{m}\right)\mathbf{e}_p + \tag{A.24}
$$

$$
-\left(\frac{k_{pd}^3}{m^2}-\frac{2k_{pp}k_{pd}}{m}+\kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\left(\frac{k_{pd}^2}{m}-k_{pp}\right)\right)\mathbf{e}_v + \tag{A.25}
$$

$$
\left. + \left(\frac{k_{pd}^2}{m^2}-\frac{k_{pp}}{m}+\frac{k_{pd}k_\Delta}{m}+k_\Delta^2+\kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\left(\frac{k_{pd}}{m}+k_\Delta\right)\right)\mathbf{f}_\Delta\right) \tag{A.26}
$$

As a consequence, it results $\dot{\boldsymbol{\omega}}_d = \boldsymbol{\omega}_{dd}$ by setting

$$
k_1 = \frac{k_{pd}^2}{m^2} - \frac{k_{pp}}{m} \tag{A.27}
$$

$$
k_2(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta) = -\left(\frac{k_{pd}^2 k_{pp}}{m^2}+\frac{k_{pp}^2}{m}+\kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\,\frac{k_{pd}k_{pp}}{m}\right) \tag{A.28}
$$

$$
k_3(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta) = -\left(\frac{k_{pd}^2 k_{pp}}{m^2}+\frac{k_{pp}^2}{m}+\kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\,\frac{k_{pd}k_{pp}}{m}\right) \tag{A.29}
$$

$$k_4(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta) = \frac{k_{pd}^2}{m^2} - \frac{k_{pp}}{m} + \frac{k_{pd}k_\Delta}{m} + k_\Delta^2 + \kappa(\mathbf{e}_p, \mathbf{e}_v, \mathbf{f}_\Delta)\ \left(\frac{k_{pd}}{m} + k_\Delta\right) \tag{A.30}$$

B

Auxiliary Material

B.1 Geometry of $SO(n)$

The rotational geometry of n-dimensional Euclidean space is characterized by the Special Orthogonal group $SO(n)$. This is a subgroup of Orthogonal group $O(n)$ such that its elements are length-preserving linear transformations whose matrix representations have unitary determinant. Formally, $SO(n) = \{\mathbf{R} \in \mathbb{R}^{n\times n} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}_n,\ \det(\mathbf{R}) = +1\}$.

B.1.1 $n = 2$

$SO(2)$, representing the space of rotations on a plane, is the group of (2×2) real orthogonal matrices with determinant $+1$, namely

$$SO(2) = \{\mathbf{R} \in \mathbb{R}^{2\times 2} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}_2,\ \det(\mathbf{R}) = +1\}. \tag{B.1}$$

To characterize the structure of the matrices belonging to $SO(2)$, let consider an orthonormal basis for $\mathbb{R}^2$, i.e., $\mathcal{B}_2 = \{[1\ \ 0]^\top, [0\ \ 1]^\top\}$. In addition, let account for an arbitrary (reference) angle $\psi \in (-\pi, +\pi]$, since a rotation in the 2D space can be fully described by a single angle. The vector $[1\ \ 0]^\top$ acts yielding

$$\mathbf{R}_\psi \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \end{bmatrix}, \tag{B.2}$$

while, rotating the other basis vector $[0\ \ 1]^\top$, it turns out

$$\mathbf{R}_\psi \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin\psi \\ \cos\psi \end{bmatrix}. \tag{B.3}$$

As a consequence, a matrix representation $\mathbf{R}_\psi$ of a ψ rotation with respect to $\mathcal{B}_2$ can be stated by using the transformed vectors as columns, i.e.,

$$\mathbf{R}_\psi = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}. \tag{B.4}$$

It can be easily proved that $SO(2)$ is abelian (commutative): for any $\mathbf{R}_{\psi_1}, \mathbf{R}_{\psi_2} \in SO(2)$, it holds that $\mathbf{R}_{\psi_1}\mathbf{R}_{\psi_2} = \mathbf{R}_{\psi_2}\mathbf{R}_{\psi_1}$. Moreover, $\mathbf{R}_{\psi_1}\mathbf{R}_{\psi_2} = \mathbf{R}_{\psi_3}$, where $\psi_3 = \psi_1 + \psi_2$. Strictly speaking, even if $\psi_1, \psi_2 \in (-\pi, +\pi]$, the resulting angle ψ_3 might be outside the parameter range, hence the angle mod 2π is always assumed.

Finally, $SO(2)$ results to be isomorphic to the circle group (in complex plane) $\mathbb{S}^1 = \{z \in \mathbb{C}\ :\ |z| = 1\}$, i.e., $SO(2) \cong \mathbb{S}^1$, or equivalently

$$\mathbb{S}^1 \ni z = e^{j\psi} \longleftrightarrow \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} = \mathbf{R}_\psi \in SO(2) \tag{B.5}$$

Given a rotation matrix $\mathbf{R}_\psi \in SO(2)$, exploiting the aforementioned isomorphism, it is possible to define its principal logarithm which results to be

$$\log(\mathbf{R}_\psi) = \begin{bmatrix} 0 & -\psi \\ \psi & 0 \end{bmatrix} \in \mathbb{R}^{2\times 2}. \tag{B.6}$$

B.1.2 $n = 3$

The orientation of each device in a multi-agent system acting in the 3D space can be represented by a matrix $\mathbf{R} \in SO(3)$, where

$$SO(3) = \left\{\mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}_3,\ \det(\mathbf{R}) = +1\right\}. \tag{B.7}$$

Each element of $SO(3)$ can be expressed as a suitable composition of the so-called *coordinate rotations*, namely the rotation matrices associated to the canonical basis of $\mathbb{R}^3$, i.e., $\mathcal{B}_3 = \{[1\ \ 0\ \ 0]^\top, [0\ \ 1\ \ 0]^\top, [0\ \ 0\ \ 1]^\top\}$. In detail, considering an angle $\psi \in (-\pi, \pi]$ and enumerating the vectors in $\mathcal{B}_3$, the coordinate rotations, $\mathbf{R}_i \in SO(3)$ for $i \in \{1, 2, 3\}$, are

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix} \mathbf{R}_2 = \begin{bmatrix} \cos\psi & 0 & \sin\psi \\ 0 & 1 & 0 \\ \sin\psi & 0 & \cos\psi \end{bmatrix} \mathbf{R}_3 = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{B.8}$$

Given this premises, it is straightforward that each rotation matrix $\mathbf{R}$ results to be associated to a certain unit vector $\mathbf{u} \in \mathbb{R}^3$ (deriving from the linear combination of the elements of $\mathcal{B}_3$) and to an angle $\theta \in (-\pi, \pi]$. Specifically, $\mathbf{R}$ denotes a rotation of θ (*rotation angle*) around $\mathbf{u}$ (*rotation axis*), according to the axis-angle representation. Exploiting the Rodrigues' rotation formula, it holds that

$$\mathbf{R} = \mathbf{I}_3 + \frac{\sin\theta}{\theta}[\mathbf{u}]_\times + \frac{1-\cos\theta}{\theta^2}[\mathbf{u}]_\times^2 = \exp\left([\mathbf{u}]_\times\right), \tag{B.9}$$

with $\theta^2 = \mathbf{u}^\top\mathbf{u}$ and $[\mathbf{u}]_\times \in \mathfrak{so}(3)$, being $\mathfrak{so}(3) = \left\{[\mathbf{u}]_\times \in \mathbb{R}^{3\times 3} \mid [\mathbf{u}]_\times = -[\mathbf{u}]_\times^\top\right\}$ the Lie algebra of $SO(3)$, namely the space of skew-symmetric matrices

Note that, differently from the group $SO(2)$, $SO(3)$ is non-abelian: the order in which rotation matrices are composed is not important. In addition, $SO(3)$ is not isomorphic to the sphere group $\mathbb{S}^2 = \{\mathbf{x} \in \mathbb{R}^3 \mid \|\mathbf{x}\|_2 = 1\}$, indeed it holds that $\mathbb{S}^2 = SO(3)\backslash SO(2)$.

To evaluate the relative orientation between two devices a metric in $SO(3)$ can be introduced. To this aim, the definition of *exponential* and *logarithm* maps in $SO(3)$ space are recalled (Ma et al. (2004)). First, given any matrix $\mathbf{R} \in SO(3)$, the *tangent space at* $\mathbf{R}$ is identified as

$$\mathfrak{T}_R(SO(3)) = \{\mathbf{R}\,[\mathbf{v}]_\times \mid [\mathbf{v}]_\times \in \mathfrak{so}(3)\}. \tag{B.10}$$

Then, the exponential map at a point $\mathbf{R} \in SO(3)$, $\exp_R(\mathbf{Q})\colon \mathfrak{T}_R(SO(3)) \to SO(3)$, is a diffeomorphism that associates to each point $\mathbf{V}$ in a neighborhood of the origin of the tangent space $\mathfrak{T}_R(SO(3))$ a point $\mathbf{S} \in SO(3)$ on the (unique) geodesic passing through $\mathbf{R}$ in the direction $\mathbf{V}$. The logarithm map $\log_R(\mathbf{S})\colon SO(3) \to \mathfrak{T}_R(SO(3))$ is instead the

Algorithm 5 Weighted rotation mean (WRM) of $\{\mathbf{R}_i\}_{i=1}^n$

```
procedure WRM($\{\mathbf{R}_i\}_{i=1}^n$,$\{w_i\}_{i=1}^n$)
    Set $\mathbf{R} := \mathbf{R}_1$ and choose a tolerance $\varepsilon > 0$.
    loop
        Compute $\mathbf{Q} = \sum_{i=1}^n w_i \log(\mathbf{R}^T\mathbf{R}_i)$
        if $\|\mathbf{Q}\| < \varepsilon$ then
            return R
        end if
        Update $\mathbf{R} = \exp(\mathbf{Q})$
    end loop
end procedure
```

inverse of the exponential map. More formally,

$$\mathbf{S} = \exp_R(\mathbf{V}) = \mathbf{R}\exp(\mathbf{R}^\top\mathbf{V}) \in SO(3), \tag{B.11}$$

$$\mathbf{V} = \log_R(\mathbf{S}) = \mathbf{R}\log(\mathbf{R}^\top\mathbf{S}) \in \mathfrak{T}_R(SO(3)). \tag{B.12}$$

Since geodesics are curves with minimum length, their length can be adopted to define the *Riemannian or geodesic distance* between the two points, namely $\mathbf{R}_i, \mathbf{R}_j \in SO(3)$. This is

$$d^2_{SO(3)}(\mathbf{R}_i, \mathbf{R}_j) = \frac{1}{2}\left\|\log\left(\mathbf{R}_i^\top\mathbf{R}_j\right)\right\|_F^2 = -\frac{1}{2}\text{trace}\left\{\log^2\left(\mathbf{R}_i^\top\mathbf{R}_j\right)\right\}, \tag{B.13}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

B.2 Averaging in $SO(3)$

There are plenty of ways to define the (weighted) mean of a set of rotations (see Hartley, Trumpf, Dai, and Li (2013)) for an extensive discussion on the subject). In this manuscript, following the work of Moakher (2002), given a set of n elements in $SO(3)$, $\{\mathbf{R}_i\}_{i=1}^n$, and a set of weights $\mathbf{w} = \{w_i\}_{i=1}^n$ such that $\sum_{i=1}^n w_i = 1$, the *weighted mean* of $\mathbf{R}_1 \dots \mathbf{R}_n$ can be computed as

$$\bar{\mathbf{R}}_\mathbf{w} = \operatorname*{arg\,min}_{\mathbf{R}\in SO(3)} \sum_{i=1}^n w_i d^2_{SO(3)}(\mathbf{R}, \mathbf{R}_i). \tag{B.14}$$

It is worth to observe that, in the case $w_i = 1/n$ for all i, the previous quantity reduces to the *simple mean* of $\{\mathbf{R}_i\}_{i=1}^n$.

A globally convergent algorithm to compute the mean of a set of rotations is presented in Manton (2004). This method consists of two main steps:

1. the computation of the mean in the tangent space,

2. its projection back onto $SO(3)$ via exponential map.

When dealing with weighted mean, this procedure can be easily generalized, as reported in Algorithm 5.

B.3 Quaternion Mathematics

The basic mathematical properties of unit quaternions are briefly summarized in the following. For a more exhaustive dissertation on this rotation representation, the reader is referred to Kuipers et al. (1999) and Diebel (2006).

A quaternion is a hyper-complex number, which is generally represented as a four dimensional vector composed by a scalar part, $\eta \in \mathbb{R}$, and a vector part, $\boldsymbol{\epsilon} \in \mathbb{R}^3$, namely

$$\mathbf{q} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix}. \tag{B.15}$$

A quaternion is said to be *unit* if it has unitary norm, i.e., $\eta^2 + \|\boldsymbol{\epsilon}\|^2 = 1$. In this case, it belongs to the unit 4D hypersphere $\mathbb{S}^3$ embedded in $\mathbb{R}^4$.

Each unit quaternion is associated to a unique rotation matrix belonging to the Special Orthogonal group $SO(3)$. Specifically, given a unit quaternion $\mathbf{q}$, the corresponding rotation matrix is computed as

$$\mathbf{R}(\mathbf{q}) = \mathbf{I}_3 + 2\eta[\boldsymbol{\epsilon}]_\times + 2[\boldsymbol{\epsilon}]_\times^2 = \mathbf{I}_3 + 2\eta[\boldsymbol{\epsilon}]_\times + 2(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top - \boldsymbol{\epsilon}^\top\boldsymbol{\epsilon}\mathbf{I}_3). \tag{B.16}$$

It can be verified that $\mathbf{R}(\mathbf{q})^\top\mathbf{R}(\mathbf{q}) = \mathbf{I}_3$ and that $\mathbf{q}_I = [1\ 0\ 0\ 0]^\top$ is an *identity* (unit) quaternion (ensuring $\mathbf{R}(\mathbf{q}_I) = \mathbf{I}_3$).

The relationship between rotation matrices and unit quaternions is not biunivocal: each rotation matrix corresponds to two unit quaternions. Indeed, it holds that $\mathbf{R}(\mathbf{q}) = \mathbf{R}(-\mathbf{q})$. This fact is justified by considering another representation for a unit quaternion, namely

$$\mathbf{q} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \mathbf{u}\sin\left(\frac{\theta}{2}\right) \end{bmatrix}, \tag{B.17}$$

where $\mathbf{u} \in \mathbb{R}^3$ denotes the rotation axis and $\theta \in (-\pi, +\pi]$ is the corresponding rotation angle as in (B.9). Note that a rotation by $-\theta$ about $-\mathbf{u}$ is represented by the same unit quaternion associated with a rotation by θ about $\mathbf{u}$ (*double coverage property*), as can be seen from (B.16).

Multiplication of two quaternions $\mathbf{q}_1, \mathbf{q}_2$ is performed by the *quaternion product*, denoted as $\odot$, which is obtained as follows

$$\mathbf{q}_1 \odot \mathbf{q}_2 = \mathbf{V}(\mathbf{q}_1)\mathbf{q}_2 = \mathbf{W}(\mathbf{q}_2)\mathbf{q}_1, \tag{B.18}$$

where

$$\mathbf{V}(\mathbf{q}) = \begin{bmatrix} \eta & -\boldsymbol{\epsilon}^\top \\ \boldsymbol{\epsilon} & \eta\mathbf{I}_3 + [\boldsymbol{\epsilon}]_\times \end{bmatrix} \quad \text{and} \quad \mathbf{W}(\mathbf{q}) = \begin{bmatrix} \eta & -\boldsymbol{\epsilon}^\top \\ \boldsymbol{\epsilon} & \eta\mathbf{I}_3 - [\boldsymbol{\epsilon}]_\times \end{bmatrix}. \tag{B.19}$$

Note that using $\mathbf{q}_3 = \mathbf{q}_1 \odot \mathbf{q}_2$, one has $\mathbf{R}(\mathbf{q}_3) = \mathbf{R}(\mathbf{q}_1)\mathbf{R}(\mathbf{q}_2)$.

Exploiting the quaternion product, it can be verified that the inverse of a given unit quaternion $\mathbf{q}$ may be chosen as

$$\mathbf{q}^{-1} = \begin{bmatrix} \eta \\ -\boldsymbol{\epsilon} \end{bmatrix}. \tag{B.20}$$

Indeed, by definition of inverse, this is $\mathbf{q}^{-1} = \mathbf{q}^*/\|\mathbf{q}\|_2 = \mathbf{q}^*$, denoting with $\mathbf{q}^*$ the conjugate of $\mathbf{q}$. Trivially, it follows that $\mathbf{q}^{-1} \odot \mathbf{q} = \mathbf{q} \odot \mathbf{q}^{-1} = 1$.

B.3.1 Quaternions Rigid Body Attitude Representation

Quaternions formalism can be used to represent the attitude of an agent belonging to a multi-device architecture and modeled as a rigid body. Let $\mathscr{F}_B$ the body frame attached to the agent so that the origin coincides with the device center of mass. The orientation of $\mathscr{F}_B$ with respect to the global inertial world frame $\mathscr{F}_W$ can be expressed through a unit quaternion $\mathbf{q}$. In detail, given a vector $\mathbf{w}$ belonging to $\mathscr{F}_B$, $\mathbf{q}$ is such that

$$\begin{bmatrix} 0 \\ \mathbf{w}' \end{bmatrix} = \mathbf{q}^{-1} \odot \begin{bmatrix} 0 \\ \mathbf{w} \end{bmatrix} \odot \mathbf{q}, \tag{B.21}$$

where vector $\mathbf{w}'$ indicates $\mathbf{w}$ in the world frame. More in general, (B.21) represents the rotation of a vector according to the quaternion convention.

Denoting by $\boldsymbol{\omega}$ the angular velocity vector in the body frame of reference, the derivative of a unit quaternion $\mathbf{q}$ is computed as

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \odot \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = \frac{1}{2}\mathbf{V}(\mathbf{q}) \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\epsilon}^\top \\ \eta \mathbf{I}_3 + [\boldsymbol{\epsilon}]_\times \end{bmatrix} \boldsymbol{\omega}. \tag{B.22}$$

Conversely, when the angular velocity is expressed in the fixed frame ($\boldsymbol{\omega}' = \mathbf{R}(\mathbf{q})\boldsymbol{\omega}$), the relation (B.22) is replaced by the following one

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega}' \end{bmatrix} \odot \mathbf{q} = \frac{1}{2}\mathbf{W}(\mathbf{q}) \begin{bmatrix} 0 \\ \boldsymbol{\omega}' \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\epsilon}^\top \\ \eta \mathbf{I}_3 - [\boldsymbol{\epsilon}]_\times \end{bmatrix} \boldsymbol{\omega}'. \tag{B.23}$$

Bibliography

Abbasi A. A. and Younis M. A survey on clustering algorithms for wireless sensor networks. *Computer communications*, 30(14):2826–2841, 2007.

Absil P.-A., Mahony R., and Sepulchre R. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

Acevedo J. J., Arrue B. C., Maza I., and Ollero A. Cooperative perimeter surveillance with a team of mobile robots under communication constraints. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5067–5072, 2013.

Adîr V. G., Stoica A. M., and Whidborne J. F. Modelling and control of a star-shaped octorotor. *Applied Mechanics and Materials*, volume 325, pages 994–998. Trans Tech Publ, 2013.

Afsar M. M. and Tayarani-N M.-H. Clustering in sensor networks: A literature survey. *Journal of Network and Computer Applications*, 46:198–226, 2014.

Aghajan H. and Cavallaro A. *Multi-camera networks: principles and applications*. Academic press, 2009.

Åkerberg J., Gidlund M., and Björkman M. Future research challenges in wireless sensor and actuator networks targeting industrial automation. *IEEE 9th International Conference on Industrial Informatics (INDIN)*, pages 410–415, 2011.

Akyildiz I. F. and Kasimoglu I. H. Wireless sensor and actor networks: research challenges. *Ad hoc networks*, 2(4):351–367, 2004.

Alberton R., Carli R., Cenedese A., and Schenato L. Multi-agent perimeter patrolling subject to mobility constraints. *American Control Conference (ACC), 2012*, pages 4498–4503, 2012.

Antonelli G., Arrichiello F., Caccavale F., and Marino A. Decentralized time-varying formation control for multi-robot systems. *The International Journal of Robotics Research*, 33(7):1029–1043, 2014.

Argentim L. M., Rezende W. C., Santos P. E., and Aguiar R. A. PID, LQR and LQR-PID on a quadcopter platform. *International Conference on Informatics, Electronics & Vision (ICIEV)*, pages 1–6, 2013.

Asimow L. and Roth B. The rigidity of graphs, II. *Journal of Mathematical Analysis and Applications*, 68(1):171–190, 1979.

Azad S., Casari P., and Zorzi M. Coastal patrol and surveillance networks using AUVs and delay-tolerant networking. *OCEANS, 2012-Yeosu*, pages 1–8, 2012.

Bajramovic F. *Self-calibration of Multi-camera Systems*. Logos Verlag Berlin GmbH, 2010.

Baseggio M., Cenedese A., Merlo P., Pozzi M., and Schenato L. Distributed perimeter patrolling and tracking for camera networks. *IEEE 49th Conference on Decision and Control (CDC)*, pages 2093–2098, 2010.

Batty M., Axhausen K. W., Giannotti F., Pozdnoukhov A., Bazzani A., Wachowicz M., Ouzounis G., and Portugali Y. Smart cities of the future. *The European Physical Journal Special Topics*, 214(1):481–518, 2012.

Bauchau O. A. and Trainelli L. The vectorial parameterization of rotation. *Nonlinear dynamics*, 32(1):71–92, 2003.

Belbachir A. N. *Smart cameras*, volume 2. Springer, 2010.

Belgioioso G., Cenedese A., and Michieletto G. Distributed partitioning strategies with visual optimization for camera network perimeter patrolling. *IEEE 55th Conference on Decision and Control (CDC)*, pages 5912–5917, 2016.

Bhanu B., Ravishankar C. V., Roy-Chowdhury A. K., Aghajan H., and Terzopoulos D. *Distributed video sensor networks*. Springer Science & Business Media, 2011.

Bianchin G., Cenedese A., Luvisotto M., and Michieletto G. Distributed fault detection in sensor networks via clustering and consensus. *IEEE 54th Annual Conference on Decision and Control (CDC)*, pages 3828–3833, 2015.

Biswas A., Guha P., Mukerjee A., and Venkatesh K. Intrusion detection and tracking with pan-tilt cameras. *International Conference on Visual Information Engineering (IET)*, pages 565–571, 2006.

Bleser G., Pastarmov Y., and Stricker D. Real-time 3D camera tracking for industrial augmented reality applications. *Journal of WSCG*, pages 47–54, 2005.

Bof N., Carli R., Cenedese A., and Schenato L. Asynchronous distributed camera network patrolling under unreliable communication. *IEEE Transactions on Automatic Control*, 2017.

Bonin-Font F., Ortiz A., and Oliver G. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263, 2008.

Borra D., Lovisari E., Carli R., Fagnani F., and Zampieri S. Autonomous calibration algorithms for networks of cameras. *American Control Conference (ACC)*, pages 5126–5131, 2012.

Bouabdallah S. and Siegwart R. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2247–2252, 2005.

Bouabdallah S. and Siegwart R. Full control of a quadrotor. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 153–158, 2007.

Bouguet J.-Y. Matlab camera calibration toolbox. *Caltech Technical Report*, 2000.

Bramberger M., Doblander A., Maier A., Rinner B., and Schwabach H. Distributed embedded smart cameras for surveillance applications. *Computer*, 39(2):68–75, 2006.

Brescianini D. and D'Andrea R. Design, modeling and control of an omni-directional aerial vehicle. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3261–3266, 2016.

Cao M., Yu C., and Anderson B. D. Formation control using range-only measurements. *Automatica*, 47(4):776–781, 2011.

Carli R., Cenedese A., and Schenato L. Distributed partitioning strategies for perimeter patrolling. *American Control Conference (ACC), 2011*, pages 4026–4031, 2011.

Carmona A., Encinas A., Gago S., Jiménez M., and Mitjana M. The inverses of some circulant matrices. *Applied Mathematics and Computation*, 270(C):785–793, November 2015.

Carrillo L. G., Dzul A., and Lozano R. Hovering quad-rotor control: A comparison of nonlinear controllers using visual feedback. *IEEE Transactions on Aerospace and Electronic Systems*, 48(4):3159–3170, 2012.

Cenedese A., Luvisotto M., and Michieletto G. Distributed clustering strategies in industrial wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 13 (1):228–237, 2017.

Chen J., Cao X., Cheng P., Xiao Y., and Sun Y. Distributed collaborative control for industrial automation with wireless sensor and actuator networks. *IEEE Transactions on Industrial Electronics*, 57(12):4219–4230, 2010.

Chen P.-Y., Yang S., and McCann J. A. Distributed real-time anomaly detection in networked industrial sensing systems. *IEEE Transactions on Industrial Electronics*, 62 (6):3832–3842, 2015.

Chevaleyre Y. Theoretical analysis of the multi-agent patrolling problem. *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT)*, pages 302–308, 2004.

Choi Y.-C. and Ahn H.-S. Nonlinear control of quadrotor for point tracking: Actual implementation and experimental tests. *IEEE/ASME Transactions on Mechatronics*, 20(3):1179–1192, 2015.

Chojnacki W. and Brooks M. Revisiting hartley's normalized eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1172–1177, 2003.

Chong C.-Y. and Kumar S. P. Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, Aug 2003.

Cook D. and Das S. K. *Smart environments: Technology, protocols and applications*, volume 43. John Wiley & Sons, 2004.

Cook D. J. Multi-agent smart environments. *Journal of Ambient Intelligence and Smart Environments*, 1(1):51–55, 2009.

Cormen T. H., Leiserson C. E., Rivest R. L., and Stein C. The algorithms of kruskal and prim. *Introduction to algorithms*, 3:631–638, 2009.

Courchay J., Dalalyan A., Keriven R., and Sturm P. A global camera network calibration method with linear programming. *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission*, 2010.

Czyzowicz J., Gasieniec L., Kosowski A., and Kranakis E. Boundary patrolling by mobile agents with distinct maximal speeds. *Algorithms–ESA 2011*, pages 701–712. Springer, 2011.

Dargie W. and Poellabauer C. *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.

Devarajan D., Radke R. J., and Chung H. Distributed metric calibration of ad hoc camera networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(3):380–403, August 2006.

Di Fava A., Satler M., and Tripicchio P. Visual navigation of mobile robots for autonomous patrolling of indoor and outdoor areas. *23th Mediterranean Conference on Control and Automation (MED)*, pages 667–674, 2015.

Di Leo G., Liguori C., Pietrosanto A., and Sommella P. A vision system for the online quality monitoring of industrial manufacturing. *Optics and Lasers in Engineering*, 89: 162–168, 2017.

Diebel J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Matrix*, 58(15-16):1–35, 2006.

Ding X., Tian Y., and Yu Y. A real-time big data gathering algorithm based on indoor wireless sensor networks for risk analysis of industrial operations. *IEEE Transactions on Industrial Informatics*, 12(3):1232–1242, 2016.

Dong X., Yu B., Shi Z., and Zhong Y. Time-varying formation control for unmanned aerial vehicles: Theories and applications. *IEEE Transactions on Control Systems Technology*, 23(1):340–348, 2015.

Du G.-X., Quan Q., Yang B., and Cai K.-Y. Controllability analysis for multirotor helicopter rotor degradation and failure. *Journal of Guidance, Control, and Dynamics*, 38(5):978–985, 2015.

Duggal V., Sukhwani M., Bipin K., Reddy G. S., and Krishna K. M. Plantation monitoring and yield estimation using autonomous quadcopter for precision agriculture. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5121–5127, 2016.

El-Hawwary M. I. and Maggiore M. Reduction theorems for stability of closed sets with application to backstepping control design. *Automatica*, 49(1):214–222, 2013.

Eren T. Using angle of arrival (bearing) information for localization in robot networks. *Turkish Journal of Electrical Engineering & Computer Sciences*, 15(2):169–186, 2007.

Eren T., Whiteley W., Belhumeur P. N., Anderson B., and others . Sensor and network topologies of formations with direction, bearing, and angle information between agents. *IEEE 42th Conference on Decision and Control (CDC)*, volume 3, pages 3064–3069, 2003.

Erginer B. and Altug E. Modeling and pd control of a quadrotor vtol vehicle. *Intelligent Vehicles Symposium, 2007 IEEE*, pages 894–899, 2007.

Eyal I., Keidar I., and Rom R. Distributed data clustering in sensor networks. *Distributed computing*, 24(5):207–222, 2011.

Fadel E., Güngör V. C., Nassef L., Akkari N., Malik M. A., Almasri S., and Akyildiz I. F. A survey on wireless sensor networks for smart grid. *Computer Communications*, 71:22–33, 2015.

Falanga D., Mueggler E., Faessler M., and Scaramuzza D. Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5774–5781, 2017.

Farina L. and Rinaldi S. *Positive linear systems: theory and applications*, volume 50. John Wiley & Sons, 2011.

Franchi A., Carli R., Bicego D., and Ryll M. Full-pose geometric tracking control on SE(3) for laterally bounded fully-actuated aerial vehicles. *arXiv preprint arXiv:1605.06645*, 2016.

Franchi A. and Giordano P. R. Decentralized control of parallel rigid formations with direction constraints and bearing measurements. *IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 5310–5317, 2012.

Franchi A. and Mallet A. Adaptive closed-loop speed control of bldc motors with applications to multi-rotor aerial vehicles. *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

Funiak S., Guestrin C., Paskin M., and Sukthankar R. Distributed localization of networked cameras. *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, pages 34–42, 2006.

Ganzha M., Jain L. C., Jarvis D., Jarvis J., and Rönnquist R. *Multiagent Systems and Applications*. Springer, 2013.

Gholami M. and Brennan R. W. Comparing two clustering-based techniques to track mobile nodes in industrial wireless sensor networks. *Journal of Systems Science and Systems Engineering*, 25(2):177–209, 2016.

Giribet J. I., Sanchez-Pena R. S., and Ghersin A. S. Analysis and design of a tilted rotor hexacopter for fault tolerance. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1555–1567, 2016.

Goebel R., Sanfelice R. G., and Teel A. R. *Hybrid Dynamical Systems: modeling, stability, and robustness*. Princeton University Press, 2012.

Gruenwedel S., Jelaca V., Niño-Castañeda J., Van Hese P., Van Cauwelaert D., Veelaert P., and Philips W. Decentralized tracking of humans using a camera network. *Proceedings of SPIE, Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*, 8301, 2012.

Gupta P. and Kumar P. R. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

Gupta S. G., Ghonge M. M., and Jawandhiya P. Review of unmanned aircraft system (UAS). *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2(4):pp–1646, 2013.

Güngör V. C. and Hancke G. P. *Industrial wireless sensor networks: Applications, protocols, and standards*. Crc Press, 2013.

Güngör V. C. and Hancke G. P. Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on Industrial Electronics*, 56 (10):4258–4265, 2009.

Haddadi S. J. and Zarafshan P. Design and fabrication of an autonomous octorotor flying robot. *3rd RSI International Conference on Robotics and Mechatronics (ICROM)*, pages 702–707, 2015.

Hancke G. P., Güngör V. C., and Hancke G. P. Guest editorial special section on industrial wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 10 (1):762–765, 2014.

Hartley R., Trumpf J., Dai Y., and Li H. Rotation averaging. *International Journal of Computer Vision*, 103(3):267–305, Jul 2013.

Hartley R. I. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

Hastie T., Tibshirani R., and Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer, 2009. ISBN 9780387848587.

Hemayed E. E. A survey of camera self-calibration. *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 351–357, 2003.

Hermann M., Pentek T., and Otto B. Design principles for industrie 4.0 scenarios. *49th Hawaii International Conference on System Sciences (HICSS)*, pages 3928–3937, 2016.

Hespanha J. P., Naghshtabrizi P., and Xu Y. A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162, Jan 2007.

Ivanov S., Bhargava K., and Donnelly W. Precision farming: Sensor analytics. *IEEE Intelligent systems*, 30(4):76–80, 2015.

Jalal A., Uddin M. Z., and Kim T.-S. Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home. *IEEE Transactions on Consumer Electronics*, 58(3), 2012.

Janai J., Güney F., Behl A., and Geiger A. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *arXiv preprint arXiv:1704.05519*, 2017.

Jiang L., Liu D.-Y., and Yang B. Smart home research. *International Conference on Machine Learning and Cybernetics*, volume 2, pages 659–663, 2004.

Karl H. and Willig A. *Protocols and architectures for wireless sensor networks*. John Wiley & Sons, 2007.

Khalil H. K. Noninear systems. *Prentice-Hall, New Jersey*, 2(5):5–1, 1996.

Kingston D., Beard R. W., and Holt R. S. Decentralized perimeter surveillance using a team of UAVs. *IEEE Transactions on Robotics*, 24(6):1394–1404, 2008.

Knuth J. and Barooah P. Collaborative 3d localization of robots from relative pose measurements using gradient descent on manifolds. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1101–1106, 2012.

Knuth J. and Barooah P. Collaborative localization with heterogeneous inter-robot measurements by riemannian optimization. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1534–1539, 2013a.

Knuth J. and Barooah P. Maximum-likelihood localization of a camera network from heterogeneous relative measurements. *American Control Conference (ACC)*, pages 2374–2379, 2013b.

Kuipers J. B. and others . *Quaternions and rotation sequences*, volume 66. Princeton university press Princeton, 1999.

Kumar R. K., Ilie A., Frahm J.-M., and Pollefeys M. Simple calibration of non-overlapping cameras with a mirror. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 2008.

Kyung C.-M. *Theory and applications of smart cameras*. Springer, 2015.

Landau H. and Odlyzko A. Bounds for eigenvalues of certain stochastic matrices. *Linear algebra and its Applications*, 38:5–15, 1981.

Lee D., Kim H. J., and Sastry S. Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. *International Journal of control, Automation and systems*, 7(3):419–428, 2009.

Lee T., Leoky M., and McClamroch N. H. Geometric tracking control of a quadrotor UAV on SE(3). *IEEE 49th Conference on Decision and Control (CDC)*, pages 5420–5425, 2010.

Leitão P., Mařík V., and Vrba P. Past, present, and future of industrial agent applications. *IEEE Transactions on Industrial Informatics*, 9(4):2360–2372, 2013.

Liang Y. and Lee H.-H. Decentralized formation control and obstacle avoidance for multiple robots with nonholonomic constraints. *American Control Conference, 2006*, pages 6–pp, 2006.

Liu T. and Jiang Z.-P. Distributed formation control of nonholonomic mobile robots without global position measurements. *Automatica*, 49(2):592–600, 2013.

Ma Y., Guo Y., Tian X., and Ghanem M. Distributed clustering-based aggregation algorithm for spatial correlated sensor networks. *IEEE Sensors Journal*, 11(3):641–648, 2011.

Ma Y., Soatto S., Kosecka J., and Sastry S. S. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2004.

Mahony R. and Kumar V. Aerial robotics and the quadrotor. *IEEE Robotics and Automation Magazine*, 19(3):19, 2012.

Manton J. H. A globally convergent numerical algorithm for computing the centre of mass on compact lie groups. *8th Control, Automation, Robotics and Vision Conference (ICARCV)*, volume 3, pages 2211–2216, 2004.

Mantzel W. E., Choi H., and Baraniuk R. G. Distributed camera network localization. *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, pages 1381 – 1386, November 2004.

Mehmood H., Nakamura T., and Johnson E. N. A maneuverability analysis of a novel hexarotor UAV concept. *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 437–446, 2016.

Mellinger D., Shomin M., Michael N., and Kumar V. Cooperative grasping and transport using multiple quadrotors. *Distributed autonomous robotic systems*, pages 545–558. Springer, 2013.

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

Micheloni C., Rinner B., and Foresti G. L. Video analysis in pan-tilt-zoom camera networks. *IEEE Signal Processing Magazine*, 27(5):78–90, 2010.

Michieletto G., Cenedese A., Zaccarian L., and Franchi A. Nonlinear control of multi-rotor aerial vehicles based on the zero-moment direction. *IFAC World Congress 2017*, pages 13686–13691, 2017a.

Michieletto G., Cenedese A., and Franchi A. Bearing rigidity theory in SE(3). *IEEE 55th Conference on Decision and Control (CDC)*, pages 5950–5955, 2016.

Michieletto G., Ryll M., and Franchi A. Control of statically hoverable multi-rotor aerial vehicles and application to rotor-failure robustness for hexarotors. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2747–2752, 2017b.

Michieletto G., Ryll M., and Franchi A. Fundamental Actuation Properties of Multi-rotors: Force-Moment Decoupling and Fail-safe Robustness. https://hal.laas.fr/hal-01612602. 2017c.

Mistler V., Benallegue A., and M'sirdi N. Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback. *Robot and Human Interactive Communication, 2001 10th IEEE International Workshop on*, pages 586–593, 2001.

Moakher M. Means and averaging in the group of rotations. *SIAM journal on matrix analysis and applications*, 24(1):1–16, 2002.

Mohamed N. and Al-Jaroodi J. A survey on service-oriented middleware for wireless sensor networks. *Service Oriented Computing and Applications*, 5(2):71–85, 2011.

Moteiv Corporation . T-mote Sky: Low power wireless sensor module, June 2006. http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf.

Nemati A. and Kumar M. Modeling and control of a single axis tilting quadcopter. *American Control Conference (ACC), 2014*, pages 3077–3082, 2014.

Neuzil J., Kreibich O., and Smid R. A distributed fault detection system based on iwsn for machine condition monitoring. *IEEE Transactions on Industrial Informatics*, 10 (2):1118–1123, 2014.

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

Olfati-Saber R. and Murray R. M. Graph rigidity and distributed formation stabilization of multi-vehicle systems. *IEEE 41st Conference on Decision and Control (CDC)*, volume 3, pages 2965–2971, 2002.

Osterlind F., Pramsten E., Roberthson D., Eriksson J., Finne N., and Voigt T. Integrating building automation systems and wireless sensor networks. *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1376–1379, 2007.

Park J. and Kim Y. Horizontal-vertical guidance of quadrotor for obstacle shape mapping. *IEEE Transactions on Aerospace and Electronic Systems*, 52(6):3024–3035, 2016.

Pasqualetti F., Zanella F., Peters J. R., Spindler M., Carli R., and Bullo F. Camera network coordination for intruder detection. *IEEE Transactions on Control Systems Technology*, 22(5):1669–1683, 2014.

Pavithra M. and Kathirvel R. Smart camera surveillance system monitoring based internet of things. *Imperial Journal of Interdisciplinary Research*, 3(6), 2017.

Pavone M., Arsie A., Frazzoli E., and Bullo F. Distributed algorithms for environment partitioning in mobile robotic networks. *IEEE Transactions on Automatic Control*, 56 (8):1834–1848, Aug 2011.

Pflugfelder R. and Bischof H. Localization and trajectory reconstruction in surveillance cameras with nonoverlapping views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):709–721, 2010.

Piovan G., Shames I., Fidan B., Bullo F., and Anderson B. D. On frame and orientation localization for relative sensing networks. *Automatica*, 49(1):206–213, 2013.

Piramuthu S. and Zhou W. *RFID and sensor network automation in the food industry: ensuring quality and safety through supply chain visibility.* John Wiley & Sons, 2016.

Potiron K., Seghrouchni A. E. F., and Taillibert P. *From fault classification to fault tolerance for multi-agent systems.* Springer, 2013.

Rahimi A., Dunagan B., and Darrell T. Simultaneous calibration and tracking with a network of non-overlapping sensors. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–187–I–194, 2004.

Rajappa S., Ryll M., Bülthoff H. H., and Franchi A. Modeling, control and design optimization for a fully-actuated hexarotor aerial vehicle with tilted propellers. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4006–4013, 2015.

Regazzoni C., Cavallaro A., Wu Y., Konrad J., and Hampapur A. Video analytics for surveillance: Theory and practice [from the guest editors]. *IEEE Signal Processing Magazine*, 27(5):16–17, 2010.

Rinner B. and Wolf W. An introduction to distributed smart cameras. *Proceedings of the IEEE*, 96(10):1565–1575, 2008.

Rodrigues R., Barreto J. P., and Nunes U. Camera pose estimation using images of planar mirror reflections. *European Conference on Computer Vision*, pages 382–395, 2010.

Romer K. and Mattern F. The design space of wireless sensor networks. *IEEE wireless communications*, 11(6):54–61, 2004.

Roy-Chowdhury A. K. and Song B. Camera networks: The acquisition and analysis of videos over wide areas. *Synthesis Lectures on Computer Vision*, 3(1):1–133, 2012.

Ryll M., Bicego D., and Franchi A. Modeling and control of FAST-hex: a fully-actuated by synchronized-tilting hexarotor. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1689–1694, 2016.

Ryll M., Bülthoff H. H., and Giordano P. R. Modeling and control of a quadrotor UAV with tilting propellers. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4606–4613, 2012.

Ryll M., Muscio G., Pierri F., Cataldi E., Antonelli G., Caccavale F., and Franchi A. 6D physical interaction with a fully actuated aerial robot. *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

Sarlette A. and Sepulchre R. Consensus optimization on manifolds. *SIAM Journal on Control and Optimization*, 48(1):56–76, 2009.

Scaramuzza D., Achtelik M. C., Doitsidis L., Friedrich F., Kosmatopoulos E., Martinelli A., Achtelik M. W., Chli M., Chatzichristofis S., Kneip L., and others . Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments. *IEEE Robotics & Automation Magazine*, 21(3):26–40, 2014.

Schaffers H., Komninos N., Pallot M., Trousse B., Nilsson M., and Oliveira A. Smart cities and the future internet: Towards cooperation frameworks for open innovation. *Future Internet Assembly*, 6656:431–446, 2011.

Schiano F., Franchi A., Zelazo D., and Giordano P. R. A rigidity-based decentralized bearing formation controller for groups of quadrotor UAVs. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

Seema A. and Reisslein M. Towards efficient wireless video sensor networks: A survey of existing node architectures and proposal for a flexi-wvsnp design. *IEEE Communications Surveys & Tutorials*, 13(3):462–486, 2011.

Shames I. and Summers T. H. Rigid network design via submodular set function optimization. *IEEE Transactions on Network Science and Engineering*, 2(3):84–96, 2015.

Sharma A. B., Golubchik L., and Govindan R. Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Transactions on Sensor Networks (TOSN)*, 6 (3):23, 2010.

Shen E. and Hornsey R. Multi-camera network calibration with a non-planar target. *IEEE Sensors Journal*, 11(10):2356–2364, 2011.

Shrouf F., Ordieres J., and Miragliotta G. Smart factories in industry 4.0: A review of the concept and of energy management approached in production based on the internet of things paradigm. *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 697–701. IEEE, 2014.

Sohraby K., Minoli D., and Znati T. *Wireless sensor networks: technology, protocols, and applications*. John Wiley & Sons, 2007.

Soro S. and Heinzelman W. A survey of visual sensor networks. *Advances in multimedia*, 2009.

Spica R., Franchi A., Oriolo G., Bülthoff H. H., and Giordano P. R. Aerial grasping of a moving target with a quadrotor UAV. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4985–4992, 2012.

Spica R. and Giordano P. R. Active decentralized scale estimation for bearing-based localization. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5084–5091, 2016.

Staub N., Mohammadi M., Bicego D., Prattichizzo D., and Franchi A. Towards robotic MAGMaS: Multiple aerial-ground manipulator systems. *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

Susto G. A., Pampuri S., Schirru A., Beghi A., and De Nicolao G. Multi-step virtual metrology for semiconductor manufacturing: A multilevel and regularization methods-based approach. *Computers & Operations Research*, 53:328–337, 2015.

Taj M. and Cavallaro A. Distributed and decentralized multicamera tracking. *Signal Processing Magazine, IEEE*, 28(3):46–58, 2011.

Trinh M. H., Park M.-C., Sun Z., Anderson B. D., Pham V. H., and Ahn H.-S. Further analysis on graph rigidity. *IEEE 55th Conference on Decision and Control (CDC)*, pages 922–927, 2016.

Tron R. *Distributed optimization on manifolds for consensus algorithms and camera network localization*. The Johns Hopkins University, 2012.

Tron R., Afsari B., and Vidal R. Riemannian consensus for manifolds with bounded curvature. *IEEE Transactions on Automatic Control*, 58(4):921–934, 2013.

Tron R. and Vidal R. Distributed image-based 3-D localization of camera sensor networks. *IEEE 48th Conference on Decision and Control held jointly with the 28th Chinese Control Conference (CDC/CCC)*, pages 901–908, 2009.

Tron R. and Vidal R. Distributed 3-D localization of camera sensor networks from 2-D image measurements. *IEEE Transactions on Automatic Control*, 59(12):3325–3340, 2014.

Turpin M., Michael N., and Kumar V. Decentralized formation control with variable shapes for aerial robots. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 23–30, 2012.

Van Rijsbergen C. J. *The geometry of information retrieval.* Cambridge University Press, 2004.

Wang S., Wan J., Li D., and Zhang C. Implementing smart factory of industrie 4.0: an outlook. *International Journal of Distributed Sensor Networks*, 12(1):3159805, 2016.

Weiser M., Gold R., and Brown J. S. The origins of ubiquitous computing research at parc in the late 1980s. *IBM Systems Journal*, 38(4):693–696, 1999.

Wooldridge M. *An introduction to multiagent systems.* John Wiley & Sons, 2009.

Wu C., Zhang Y., Sheng W., and Kanchi S. Rigidity guided localisation for mobile robotic sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 6(2):114–128, 2010.

Zanella A., Bui N., Castellani A., Vangelista L., and Zorzi M. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1):22–32, 2014.

Zelazo D., Franchi A., Allgöwer F., Bülthoff H. H., and Giordano P. R. Rigidity maintenance control for multi-robot systems. *Robotics: science and systems*, pages 473–480, 2012.

Zelazo D., Franchi A., Bülthoff H. H., and Giordano P. R. Decentralized rigidity maintenance control with range measurements for multi-robot systems. *The International Journal of Robotics Research*, 34(1):105–128, 2015a.

Zelazo D., Franchi A., and Giordano P. R. Rigidity theory in SE(2) for unscaled relative position estimation using only bearing measurements. *European Control Conference (ECC)*, pages 2703–2708, 2014.

Zelazo D., Giordano P. R., and Franchi A. Bearing-only formation control using an SE(2) rigidity theory. *IEEE 54th Annual Conference on Decision and Control (CDC)*, pages 6121–6126, 2015b.

Zhang W., Han G., Feng Y., Cheng L., Zhang D., Tan X., and Fu L. A novel method for node fault detection based on clustering in industrial wireless sensor networks. *International Journal of Distributed Sensor Networks*, 11(7):230521, 2015.

Zhao S. and Zelazo D. Bearing rigidity and almost global bearing-only formation stabilization. *IEEE Transactions on Automatic Control*, 61(5):1255–1268, 2016.

Zhao Z. and Liu Y. Practical multi-camera calibration algorithm with 1D objects for virtual environments. *IEEE International Conference on Multimedia and Expo*, pages 1197–1200, 2008.