Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Master Thesis

Distributed Coordination of Quadcopters Swarms via Online Feedback Equilibrium Seeking

Andrea Da Col
June 11, 2023

Advisors
Giuseppe Belgioioso
Ezzat Elokda
Florian Dörfler

Contents

Preface

My utmost gratitude goes to my supervisors, Giuseppe Belgioioso and Ezzat Elokda, who assisted me throughout the duration of this project.

Abstract

Integration of unmanned aerial vehicles (UAVs) as wireless relays in traditional telecommunication infrastructures has proved to be effective in mitigating network congestion, enhancing Internet services, and establishing ad-hoc networks in complicated environments. Efficient and scalable coordination mechanisms are crucial for managing and maintaining optimal, stable UAV formations, and adapting to the rapidly changing network demands.
In this thesis, two fully-distributed control architectures are proposed that are based on Feedback Equilibrium Seeking [1], a unifying framework for designing algorithmic-based output feedback controllers. In particular, algorithmic trajectory planners are proposed that dynamically steer UAV swarms near efficient network configurations that give the best *quality of service* (QoS) to a group of Internet subscribers. Robustness to changes in the Internet demand for the discrete-time planner in closed-loop with a pre-stabilized UAV network is studied in a sampled-data setting. We provide a formal robust stability analysis of the obtained sampled-data system to ensure tracking of an optimal (time-varying, a-priori unknown) swarm configuration and experimentally validate the theoretical findings using real-world quadcopters. Precise convergence of a fleet of Bitcraze Crazyflies to the desired operating conditions is achieved in the presence of exogenous disturbances affecting the users' request for Internet services, thus showcasing effectiveness of the adopted framework in modelling relevant objectives for mobile wireless networks.

List of Figures

Chapter 1

Introduction

Improved maneuverability and accessibility, as well as low deployment costs, motivate the increasing research interest towards UAVs, which are expected to play a big role in the upcoming technological developments [2]. A considerable push in this direction comes from the telecommunication sector. In fact, the integration of UAV-aided networks into more traditional infrastructures could enable reliable wireless communication paths between distant transceivers, and data offloading in case of congestion. Use of intelligent agents having tight requirements on the quality of incoming real-time data-flows (e.g., autonomous vehicles) would also benefit from an augmentation of the overall network capacity.

Motivated by the recent outbreak of the fifth generation of wireless networks (5G) and the Internet of Things [3], this work addresses dynamic enhancement of existing ground networks in terms of throughput, capacity and optimal customer coverage. In particular, the focus is put on the synthesis of distributed controllers to perform optimal *formation* and *coverage control* of multi-robot systems in wireless networks. Note that this classification is general and that it includes UAV-assisted wireless relays [4, 5] and sensor networks [6, 7, 8], as well as search and rescue missions [9] and mobile ad-hoc networks for traffic management of autonomous vehicles [10] as special cases. The words *formation control* and *optimal coverage* will be used consistently throughout this paper. In particular, with *formation control* we mean convergence of many interconnected agents to a desired network configuration. Practical examples include organization in ad-hoc infrastructures for connection of distant users and quick recovery from natural disasters. In the attempt of jointly optimizing quality of inter-agent data exchange and energy consumption, we will consider the length of the established wireless links as a metric to design the performance cost. The latter is more general than *formation control* as it extends to time-varying, self-arranging graph topologies. Both will be considered to take place in rapidly changing environments to show adaptability and robust stability of the proposed feedback controller. Throughout this thesis, we will use the words UAV, vehicle, robot and agent interchangeably.

1.1 Related Works

Recent advances in consensus-based and game-theoretic formation control [11] have motivated the research interest toward many possible applications of wireless robotic networks, both terrestrial and aerial. Channel-aware path planning for optimizing energy consumption of a network of unicycles is investigated in [12] in realistic wireless setting [13]. In [6, 14] distributed gradient-based algorithms are discussed for multi-agent systems, that steer a network to the optimal solutions of *optimal coverage*. Properties and extensions of such algorithms are presented in [15, 16]. A first attempt to use feedback-based optimization [17] for multi-robot coordination is

proposed in [18] to study convergence of a network of differential drives to a reference formation. The more recent literature has started to move towards UAV-enabled wireless networks. A thorough survey on this topic is found in [3]. Optimal UAV placement for network-wide information exchange and computationally light algorithm implementations are addressed in [9]. In [19, 20] optimal placement for small networks of drones is formulated using non-cooperative games that must provide backbone communication channels to ground users. Evolutionary games under time-varying communication topologies are introduced in [21] to optimally solve UAV coordination tasks in a distributed fashion. Finally, distributed algorithms for path planning and end-to-end throughput optimization are proposed in [22, 23]. The last two references however don't consider a game-theoretical approach.

1.2 Statement of Contribution

Our contributions are threefold. First, we tailor FES for quadcopters swarms coordination. In doing that, we show that *formation control* and *optimal coverage* are encompassed by the theoretical results in [1] and recognise FES as a valid framework for modelling meaningful objectives for mobile-base robotic networks. Second, we design two distributed, scalable and robust output feedback controllers for the two aforementioned control problems. Local input-to-state stability of the closed-loop system is proved by showing compliance to some non-stringent requirements on the pre-stabilized physical systems (i.e., the locally controlled quadcopters), strong regularity of the GE and robust converegence of the ES algorithm. Lastly, validation of the theoretical finding is done on a fleet of real-world quadcopters. In particular, we verify convergence to efficient steady states for sampling periods that are small enough. From the best of our knowledge, this is the first attempt to experimentally validate FES in multi-agent robotics, as most works only provide simulations as running examples.

1.3 Report Structure

The remainder of this report is organized as follows. A detailed problem formulation for optimal *formation* and *coverage control* is found in Chapter 2. Later chapters will discuss the sufficient conditions given in Section 1.4 for local input-to-state stability of the feedback interconnection of the physical plant with the ES algorithm. In Chapter 3 we investigate local stabilizability of the non-linear model of a quadcopter using both LQR and MPC control policies. Our discussion is limited to a disturbance-free scenario. Chapter 4 addresses the design of the optimal control problem in terms of GEs and the use of two common discrete-time algorithms found in literature. Experimental results are presented in Chapter 5 to substantiate the theoretical findings. Viable directions for future research are summarized in Chapter 6 to conclude the report.

1.4 Preliminaries

1.4.1 Notation

We will denote by $\mathbb{R}_{++}$ ($\mathbb{R}_{+}$) the set of positive (non-negative) real numbers. Let $I_n \in \mathbb{R}^{n\times n}$ be the identity matrix of dimension $n \in \mathbb{N}_{++}$; $A \otimes B \in \mathbb{R}^{(m_b m_a)\times(n_a n_b)}$ be the Kronecker product of the matrices $A \in \mathbb{R}^{m_a \times n_a}$, $B \in \mathbb{R}^{m_b \times n_b}$. The set of positive (semi-) definite matrices of $\mathbb{R}^{n\times n}$ is denoted by $\mathbb{S}^n_{++}$ ($\mathbb{S}^n_{+}$). Given $P \in \mathbb{S}^n_{++}$, $\boldsymbol{x} \in \mathbb{R}^n$, $\|\boldsymbol{x}\|_P \coloneqq \sqrt{\boldsymbol{x}^\top P \boldsymbol{x}}$ is a weighted norm. We use $\|\boldsymbol{x}\|$ to indicate the Euclidean norm $\|\boldsymbol{x}\|_I$ of $\mathbb{R}^n$. The linear space of continuous-time signals is denoted by $\mathbb{L}^n = \{f \colon \mathbb{R}_+ \to \mathbb{R}^n\}$. Given $\boldsymbol{x} \in \mathbb{L}^{n_x}$, we denote the signal norm as $|\boldsymbol{x}| = \overline{\sup}_{t\in\mathbb{R}_+} \|\boldsymbol{x}(t)\|_I$. A function $\gamma \colon \mathbb{R}_+ \to \mathbb{R}_+$ is of class $\mathcal{K}$ if it is continuous, strictly increasing, and such that $\gamma(0) = 0$.

If it is also unbounded, then $\gamma \in \mathcal{K}_\infty$. By converse, a function $\sigma\colon \mathbb{R}_+ \to \mathbb{R}_+$ is of class $\mathcal{L}$ if it is continuous, strictly decreasing and its limit at infinity is zero. Given a closed convex set $\mathcal{X} \subseteq \mathbb{R}^n$, let $\iota_{\mathcal{X}}\colon \mathbb{R}^n \to \{0, \infty\}$ be the indicator function on $\mathcal{X}$, $\mathcal{N}_{\mathcal{X}}\colon \mathcal{X} \rightrightarrows \mathbb{R}^n$ the sub-differential of $\iota_{\mathcal{X}}$, and $\Pi_{\mathcal{X}}\colon \mathbb{R}^n \to \mathcal{X}$ the orthogonal projection onto $\mathcal{X}$.

1.4.2 Feedback Equilibrium Seeking

Feedback Equilibrium Seeking (FES) is a general framework to design output feedback controllers that maintain complex dynamical systems near efficient (a-priori unknown) operating conditions. Following a *stabilize-then-optimize* approach, FES-based controllers are obtained by the direct implementation of a discrete-time equilibrium seeking (ES) algorithm in closed-loop with a locally pre-stabilized continuous-time plant. A thorough review of this subject, as well as a complete stability analysis of the sampled-data, closed-loop interconnection is found in [1]. In what follows, we go through the main idea of FES control and the sufficient conditions that ensure local input-to-state stability of the sampled-data system. Our discussion is given a modular structure that is reflected by later chapters.

Consider the general non-linear, dynamical system

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)), \tag{1.1a}$$
$$\boldsymbol{y}(t) = g(\boldsymbol{x}(t), \boldsymbol{w}(t)), \tag{1.1b}$$

where $\boldsymbol{x} \in \mathbb{L}^{N_x}$, $\boldsymbol{u} \in \mathbb{L}^{N_u}$, $\boldsymbol{w} \in \mathbb{L}^{N_w}$, $\boldsymbol{y} \in \mathbb{L}^{N_y}$ denote the state, the control input, the external disturbances and the output respectively, and $\boldsymbol{u}(t) \in \mathcal{U}$, $\boldsymbol{w}(t) \in \mathcal{W}$ for all $t \in \mathbb{R}_+$.
Let $p\colon \mathcal{U} \times \mathcal{W} \to \mathbb{R}^{N_x}$ such that $f(p(\boldsymbol{u}, \boldsymbol{w}), \boldsymbol{u}, \boldsymbol{w}) = 0$ be the steady-state map and $h\colon \mathcal{U} \times \mathcal{W} \to \mathbb{R}^{N_y}$ the steady-state input-output map. Namely, p maps the control inputs and disturbances to an equilibrium point of (1.1a), while h yields the output at equilibrium. The following assumption ensures that the sensitivity mappings and a solution to (1.1) exist and are well defined.

Assumption 1 (Robust Stability of the Plant)**.** *The following specifications, if verified, ensure that the solution trajectories and the steady state mappings of the plant are well defined:*

(i) *f is locally Lipschitz continuous,*

(ii) *g is globally L_g-Lipschitz continuous,*

(iii) *$\mathcal{W}$ and $\mathcal{U}$ are compact and convex,*

(iv) *(1.1) is LISS, i.e., there exist $\varepsilon_x, \varepsilon_w, \alpha_1, \alpha_2, \alpha_3 > 0$, a continuously differentiable function V, and $\sigma_1 \in \mathcal{K}$ such that for any constant $\boldsymbol{u} \in \mathcal{U}$*

$$\alpha_1 \|\boldsymbol{x} - p(\boldsymbol{u}, \boldsymbol{w})\|^2 \leq V(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \leq \alpha_2 \|\boldsymbol{x} - p(\boldsymbol{u}, \boldsymbol{w})\|^2,$$
$$\dot{V}(\boldsymbol{x}(t), \boldsymbol{u}, \boldsymbol{w}(t)) \leq -\alpha_3 V(\boldsymbol{x}(t), \boldsymbol{u}, \boldsymbol{w}(t)) + \sigma_1(\|\dot{\boldsymbol{w}}(t)\|),$$

if $V(\boldsymbol{x}(t), \boldsymbol{u}, \boldsymbol{w}(t)) \leq \varepsilon_x$ and $\|\dot{\boldsymbol{w}}(t)\| \leq \varepsilon_w$.

We invite the reader to interpret $\boldsymbol{u}$ as a variable upon which we define a setpoint for (1.1). Optimality of $\boldsymbol{u}$ is expressed using the parametric GE:

$$0 \in G(\boldsymbol{z}, \boldsymbol{s}) + \mathcal{A}(\boldsymbol{z}), \tag{1.2a}$$
$$\boldsymbol{s} = h(\boldsymbol{u}, \boldsymbol{w}), \tag{1.2b}$$
$$\boldsymbol{u} = q(\boldsymbol{z}), \tag{1.2c}$$

where $\boldsymbol{z} \in \mathbb{R}^{N_z}$ is the state variable of the FES-controller and $\boldsymbol{s} \in \mathbb{R}^{N_y}$ is the output of (1.1) at steady state. In particular, (1.2a) specifies the first-order conditions for optimality through the single-valued map $G\colon \mathbb{R}^{N_z} \times \mathbb{R}^{N_w} \to \mathbb{R}^{N_z}$ and the set-valued mapping $\mathcal{A}\colon \mathbb{R}^{N_z} \rightrightarrows \mathbb{R}^{N_z}$, (1.2b) is the output sensitivity and (1.2c) maps the controller state to the control input of the pre-stabilized system. (1.2) can be rewritten more compactly as

$$0 \in \mathbb{G}(\boldsymbol{z}, \boldsymbol{w}) + \mathcal{A}(\boldsymbol{z}), \tag{1.3}$$

where $\mathbb{G}(\boldsymbol{z}, \boldsymbol{w}) = G(\boldsymbol{z}, h(q(\boldsymbol{z}), \boldsymbol{w}))$. Let $S\colon \mathcal{W} \to \mathbb{R}^{N_z}$ denote the disturbance-to-solution mapping, which is defined as

$$S(\boldsymbol{w}) = \{\boldsymbol{z} \in \mathbb{R}^{N_z} \mid 0 \in \mathbb{G}(\boldsymbol{z}, \boldsymbol{w}) + \mathcal{A}(\boldsymbol{z})\},$$

and $\mathcal{S}$ be the set of solution trajectories, i.e.,

$$\mathcal{S}(\boldsymbol{w}) = \{\boldsymbol{z} \in \mathbb{L}^{N_z} \mid \boldsymbol{z}(t) \in S(\boldsymbol{w}(t)),\ \forall t \in \mathbb{R}_+\}, \tag{1.4}$$

where $\boldsymbol{w} \in \mathbb{L}^{N_w}$. The following assumption is made to ensure that (1.4) contains finitely many isolated elements, and thus (1.3) is well posed:

Assumption 2 (Strong Regularity of the GE)**.** *The solution-tracking problem is well defined if:*

(i) *q is globally L_q-Lipschitz,*

(ii) *G is continuously differentiable,*

(iii) *the mapping $\mathbb{G}(\cdot, \boldsymbol{w}) + \mathcal{A}(\cdot)$ is strongly regular at all points $\boldsymbol{z} \in S(\boldsymbol{w})$ for all $\boldsymbol{w} \in \mathcal{W}$.*

Inability to measure the disturbance, or simply incomplete knowledge of it, as well as impossibility to write the solution mapping in close form, make it difficult to compute the optimal control input trajectory $\boldsymbol{u}^* \in \mathbb{L}^{N_u}$. In FES, the solution of a GE is found using ES algorithms that incorporate online measurements $\boldsymbol{y}^k$ from the plant instead of the steady-state output h, which might be unavailable. The following update rule is modified:

$$\boldsymbol{s}^k = h(q(\boldsymbol{z}^k), \boldsymbol{w}), \tag{1.5a}$$

$$\boldsymbol{z}^{k+1} = T(\boldsymbol{z}^k, \boldsymbol{s}^k), \tag{1.5b}$$

where $T\colon \mathbb{R}^{N_z} \times \mathbb{R}^{N_y} \to \mathbb{R}^{N_z}$. (1.5) can be recast in the more compact parametric rule, i.e.,

$$\mathbb{T}(\boldsymbol{z}, \boldsymbol{w}) = T(\boldsymbol{z}, h(q(\boldsymbol{z}), \boldsymbol{w})). \tag{1.6}$$

which is ensured to be locally convergent if the following specifications are met.

Assumption 3 (Robust Convergence of the Algorithm)**.** *The following statements hold:*

(i) *for all $\boldsymbol{w} \in \mathcal{W}$, $\boldsymbol{z} = T(\boldsymbol{z}, \boldsymbol{w})$ if and only if $\boldsymbol{z} \in S(\boldsymbol{w})$,*

(ii) *there exists a continuous function $W\colon \mathbb{R}^{N_z} \times \mathcal{W} \to \mathbb{R}$, a constant $\varepsilon > 0$, $\alpha_4, \alpha_5 \in \mathcal{K}_\infty$, and $\alpha_6 \in \mathcal{K}$ such that for all $\boldsymbol{w} \in \mathcal{W}$, $\overline{\boldsymbol{z}} \in S(\boldsymbol{w})$ and $\boldsymbol{z} \in \{\boldsymbol{\xi} \mid \|\boldsymbol{\xi} - \boldsymbol{z}^*\| \leq \varepsilon\}$*

$$\alpha_4(\|\boldsymbol{z} - \overline{\boldsymbol{z}}\|) \leq W(\boldsymbol{z}, \boldsymbol{w}) \leq \alpha_5(\|\boldsymbol{z} - \overline{\boldsymbol{z}}\|),$$
$$W(\mathbb{T}(\boldsymbol{z}, \boldsymbol{w}), \boldsymbol{w}) \leq W(\boldsymbol{z}, \boldsymbol{w}) - \alpha_6(\|\boldsymbol{z} - \overline{\boldsymbol{z}}\|),$$

(iii) *for all fixed $\boldsymbol{z}$ there exists a constant L_T such that*

$$\|T(\boldsymbol{z}, \boldsymbol{s}) - T(\boldsymbol{z}, \boldsymbol{s}')\| \leq L_T \|\boldsymbol{y} - \boldsymbol{y}'\|, \quad \forall \boldsymbol{s}, \boldsymbol{s}' \in \mathbb{R}^{N_y}.$$

Among the main contributions of [1] we recognise the identification of sufficient conditions for convergence of multi-agent cyber-physical systems with potentially self-interested agents to strategic steady-state trajectories in a sampled-data setting, as well as a considerable extension of [24] to first- and second-order, locally convergent discrete-time algorithms for non-monotone GEs.

1.4.3 Centroidal Voronoi Partitions

Let $\mathcal{I} = \{1, \dots, N\}$, $\mathcal{D} \in \mathbb{R}^n$ be a compact polytope, $\boldsymbol{d} = (\boldsymbol{d}_1^\top, \dots, \boldsymbol{d}_N^\top)^\top$ a collection of points such that $\boldsymbol{d}_i \in \mathcal{D}$ for all $i \in \mathcal{I}$. Define now a partition $\mathfrak{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_N\}$ of $\mathcal{D}$ in N polytopes having disjoint interiors and satisfying $\mathcal{D} = \bigcup_{i \in \mathcal{I}} \mathcal{D}_i$. We refer to one $\mathcal{D}_i$ as a *dominance region* for $\boldsymbol{d}_i$ and call two partitions equivalent if all their elements only differ by a set of measure zero. In addition, we let $\varrho \colon \mathcal{D} \to \mathbb{R}_+$ be a probability measure expressing the likelihood of a certain random event on $\mathcal{D}$ and $d \colon \mathbb{R}_+ \to \mathbb{R}_+$ be an increasing differentiable function of a distance metric [6, 25]. The problem of *optimal coverage* is structured as

$$\tilde{\mathbb{J}}\left(\boldsymbol{d}, \mathfrak{D}(\boldsymbol{d})\right) = \sum_{i \in \mathcal{I}} \int_{\mathcal{D}_i} d(\|\boldsymbol{d}_i - \boldsymbol{q}\|) \varrho(\boldsymbol{q}) d\boldsymbol{q} \tag{1.7}$$

and evaluates the overall performance of agents of $\mathcal{I}$ to provide a service in their *dominance region*. Optimization of (1.7) considers two aspects simultaneously, namely the choice of a configuration of locations $\boldsymbol{d} \in \bigtimes_{i \in \mathcal{I}} \mathcal{D}$ and of the associated partition $\mathfrak{D}$.
With this in mind we introduce known results on *centroidal Voronoi tessellations* (CVTs), and refer the reader to [25, 26] for more details. The Voronoi region generated by $\boldsymbol{d}_i$ is the set

$$\mathcal{R}_i = \{\boldsymbol{d}_i \in \mathcal{D} \mid \|\boldsymbol{d}_i - \boldsymbol{q}\| \leq \|\boldsymbol{d}_j - \boldsymbol{q}\|, \forall j \in \mathcal{I}, i \neq j\}. \tag{1.8}$$

where we ensure non-degeneracy of $\mathcal{R}_i$ by allowing the generators to stay inside $\Gamma = \{\boldsymbol{d} \in \mathbb{R}^{Nn} \mid \boldsymbol{d}_i \neq \boldsymbol{d}_j, \forall j \in \mathcal{I} \setminus i, \boldsymbol{z}_i \in \mathcal{D}, \forall i \in \mathcal{I}\}$ [16]. The collection of the Voronoi regions induced by the points in $\boldsymbol{d}$ is called the Voronoi diagram generated by $\boldsymbol{d}$ and is denoted with $\mathfrak{R}$. Two sites $i, j \in \mathcal{I}$ such that $i \neq j$ are said to be adjacent if $\mathcal{R}_i \cap \mathcal{R}_j \neq \varnothing$ and has dimension $n - 1$. We denote with $\mathcal{N}_i$ the set of points that are adjacent to i. We are considering the 2-norm in our discussion, but invite to note that the choice of the metric in (1.8) is really application dependent [27]. It is easy to check that every $\mathcal{R}_i$ defined using the Euclidean distance can be obtained as the intersection of $\mathcal{D}$ with

$$\mathcal{H}_i^j = \{\boldsymbol{q} \in \mathcal{D} \mid (\boldsymbol{d}_j - \boldsymbol{d}_i) \cdot (2\boldsymbol{q} - \boldsymbol{d}_i - \boldsymbol{d}_j) \leq 0\}, \quad \forall j \in \mathcal{I} \setminus i. \tag{1.9}$$

This is typically known as the orthogonal bisector property, and shows that all Voronoi cells of this type are compact polytopes. We can use the above definitions to present key results for *optimal coverage* on non-degeneracy of the local optimizers.

Lemma 1 (Existence of Minimizers [25])**.** *Given the region $\mathcal{D} \in \mathbb{R}^n$, a density function $\varrho \colon \mathcal{D} \to \mathbb{R}_+$, any collection of N points belonging $\mathcal{D}$ called $\boldsymbol{d} = [\boldsymbol{d}_i]_{i \in \mathcal{I}}$ and a partition $\mathfrak{D}$ of $\mathcal{D}$, a necessary condition for (1.7) to be minimized is that $\mathfrak{D}$ is the Voronoi diagram generated by $\boldsymbol{d}$ and each $\boldsymbol{d}_i$ is the center of mass of $\mathcal{R}_i$.*

A Voronoi diagram generated by $\boldsymbol{d}$ satisfying Lemma 1 is called a *centroidal Voronoi diagram*, and is generally not unique.

Remark 1 (Multiple Minimizers [6])**.** *Consider the vector $\hat{\boldsymbol{d}} = [\hat{\boldsymbol{d}}_i]_{i \in \mathcal{I}}$ of points belonging to $\mathcal{D}$, an associated partition $\hat{\mathfrak{D}}$ and the operation of vector permutation $p : \mathbb{R}^{Nn} \to \mathbb{R}^{Nn}$, $p \in \mathcal{P}$ with $\mathcal{P}$ the set of all possible permutations. The pairs $(\hat{\boldsymbol{d}}, \hat{\mathfrak{D}})$ and $(p(\hat{\boldsymbol{d}}), p(\hat{\mathfrak{D}}))$ achieve the same value for (1.7) for all $p \in \mathcal{P}$.*

Consider now the minimizers obtained permutation of a collection of vehicles' locations: the cost functional will still admit multiple CVTs (i.e., stationary points) in $\mathbb{R}^{Nn}$, if $n \geq 2$. Existence of a unique global minimum has been proved in [25] only for $\mathcal{D} \subset \mathbb{R}$ and strictly log-concave densities.

Lemma 2 (Properties of Local Minimizers [25])**.** *If ϱ is positive except on a subset of $\mathcal{D}$ of measure zero, then local minimizers of (1.7) are such that $\boldsymbol{z}_i \neq \boldsymbol{z}_j$ for all $j \in \mathcal{I} \setminus i$.*

1.4.4 Lloyd Map

Let $\boldsymbol{e} = (\boldsymbol{e}_1^\top, \dots, \boldsymbol{e}_H^\top)^\top$ denote the vector containing the vertices of $\mathcal{R}_i$ for all $i \in \mathcal{I}$ without repetitions, and $\mathfrak{T}_{\boldsymbol{p}}$ the Delaunay triangulation of $\boldsymbol{p}$, i.e., the dual graph of the Voronoi partition $\mathfrak{R}$ induced by $\boldsymbol{p}$. Each entry of $\boldsymbol{e}$ can be obtained as the circumcenter of the hypersphere generated by points that are adjacent in the sense of Voronoi. We denote with $E : \mathbb{R}^{Nn} \to \mathbb{R}^{Hn}$ the function that maps the generators of $\mathfrak{R}$ to its vertices, and with $\boldsymbol{e}^{\mathcal{R}_i}$ the vector containing all the vertices of $\mathcal{R}_i$. The latter is obtained from $\boldsymbol{e}^{\mathcal{R}_i} = \Xi^{\mathcal{R}_i}\boldsymbol{e}$, where $\Xi^{\mathcal{R}_i}$ is a binary matrix of appropriate dimensions that extracts the H_i vertices of $\mathcal{R}_i$ from $\boldsymbol{e}$. Note that each $\mathcal{R}_i$ that is defined using the 2-norm in (1.8) can be expressed as the convex combination of its vertices, i.e.,

$$\mathcal{R}_i = \left\{ \boldsymbol{q} \in \mathcal{D} \,\middle|\, \boldsymbol{q} = \boldsymbol{e}_{H_i}^{\mathcal{R}_i} + \sum_{j=1}^{H_i-1} \lambda_j \left(\boldsymbol{e}_j^{\mathcal{R}_i} - \boldsymbol{e}_{H_i}^{\mathcal{R}_i} \right) \right\}, \tag{1.10}$$

where $\boldsymbol{\lambda} = \left(\lambda_1, \dots, \lambda_{H_i-1}\right)^\top$ belongs to the simplex of dimension $H_i - 1$.
Let also $F \colon \mathbb{R}^{Hn} \to \mathbb{R}^{Nn}$ map the vertices of $\mathfrak{R}$ to the centers of mass of the Voronoi regions:

$$F \colon \boldsymbol{e} \to \left[\frac{\int_{\mathcal{R}_i(\boldsymbol{e})} \boldsymbol{q}\varrho(\boldsymbol{q})d\boldsymbol{q}}{\int_{\mathcal{R}_i(\boldsymbol{e})} \varrho(\boldsymbol{q})d\boldsymbol{q}} \right]_{i \in \mathcal{I}}. \tag{1.11}$$

The Lloyd map is then defined as $LL = F \circ E$ and turns the generators of $\mathfrak{R}$ into the centers of mass of its Voronoi regions, i.e.,

$$LL \colon \boldsymbol{d} \to \left[\frac{\int_{\mathcal{R}_i(E(\boldsymbol{d}))} \boldsymbol{q}\varrho(\boldsymbol{q})d\boldsymbol{q}}{\int_{\mathcal{R}_i(E(\boldsymbol{d}))} \varrho(\boldsymbol{q})d\boldsymbol{q}} \right]_{i \in \mathcal{I}}. \tag{1.12}$$

Chapter 2

Problem Statement

Let $\mathcal{I} = \{1, \ldots, N\}$ be a set of intelligent, interconnected agents that mutually exchange information using a certain wireless communication protocol. Inter-agent communication is assumed to be synchronous, instantaneous and accurate. To model bidirectional exchange of data, an undirected and potentially time-varying graph topology $\mathcal{G}_{\mathcal{I}}(\mathcal{I}, \mathcal{E}_{\mathcal{I}})$ is considered, where $\mathcal{E}_{\mathcal{I}} \subseteq \mathcal{I} \times \mathcal{I}$ is a collection of unordered pairs of nodes $\{i, j\}$, called edges, that model wireless connection between two distinct agents labelled by $i, j \in \mathcal{I}$. Every $i \in \mathcal{I}$ is located inside $\mathcal{FR}$, where $\mathcal{FR} \subset \mathbb{R}^3$ is an admissible flying region. We will assume connectivity of $\mathcal{G}_{\mathcal{I}}$, namely existence of a path connecting any pair of nodes. Indirectly, $\mathcal{G}_{\mathcal{I}}$ models availability of partial knowledge of the full network's state: interaction between neighbouring agents is promoted as a way to access global information. This, as well as limited computational resources, motivate the use of a distributed approach to the multi-agent control task. In practice, one agent chooses the control action depending on the information it possesses, which is obtained from its neighbours.

In our problem formulation $\mathcal{I}$ denotes a swarm of quadcopters each having non-linear dynamics, $\mathcal{FR}$ is the compact polytope in which they are allowed to fly, and $\mathcal{G}_i$ specifies the wireless links through which inter-UAV communication takes place. The collective state of the multi-agent system is denoted by $\boldsymbol{x} \in \mathbb{R}^{N_x}$ obtained by stacking the individual state vectors of the quadcopters. The same procedure is done for the control input vector $\boldsymbol{u} \in \mathcal{U}$, where $\mathcal{U} = \bigtimes_{i \in \mathcal{I}} \mathcal{U}_i$ is the space of admissible network configurations, and $\mathcal{U}_i = \mathcal{FR}$ is the set of reference positions for the i-th quadcopter. In the following section, we provide the dynamic equations of a (generic) quadcopter and drop the label i for convenience.

2.1 Equations of Motion

This section focuses on modelling the full non-linear equations of motion (EoMs) of one quadcopter and constitutes the first step towards the stability analysis of a formation of such agents. Especially, our interest in considering non-linear EoMs is twofold as they serve for the later synthesis of local control laws, as well as a tool for capturing the generality of the FES paradigm. Note that our discussion on quadcopter's dynamics does not aim at being complete, but rather at listing the key concepts that simplify the control task. For more details see [28, 29, 30, 31].

A quadcopter is an under-actuated aircraft often modelled as a rigid body moving in space, mounting co-planar propellers that can apply a vertical thrust force. In our discussion, it will be assumed to be symmetric about its centre-of-mass (CoG), with equidistant propellers.
Let us define an inertial frame and a body-fixed frame, which we denote as $\mathcal{O}$ and $\mathcal{B}$ respectively. The first has the z-coordinate axis aligned with the gravity vector and facing upwards, the latter is rigidly attached to the body of the quadcopter, has the origin fixed at its CoG and

the z-axis aligned with the direction of positive thrust. The position of $\mathcal{B}$ relative to $\mathcal{O}$ is the vector $\boldsymbol{p}^{\mathcal{O}} = (p_x^{\mathcal{O}}\ p_y^{\mathcal{O}}\ p_z^{\mathcal{O}})$ pointing from the origin of $\mathcal{O}$ to the CoG. Rotations of $\mathcal{B}$ with respect to $\mathcal{O}$ are expressed using the intrinsic ZYX Euler angles convention. The vector of Euler angles $\boldsymbol{\psi} = (\gamma\ \beta\ \alpha)$ is obtained stacking the roll, pitch and yaw values and describes the linear transformation ${}_{\mathcal{O}}R^{\mathcal{B}}$ from $\mathcal{B}$ to $\mathcal{O}$. The vectors $\boldsymbol{p}^{\mathcal{O}}$ and $\boldsymbol{\psi}$ express the configuration of a rigid body in $\mathbb{R}^3$ without ambiguity. To conclude, we introduce the linear velocity vector $\dot{\boldsymbol{p}}^{\mathcal{O}} = (\dot{p}_x^{\mathcal{O}}\ \dot{p}_y^{\mathcal{O}}\ \dot{p}_z^{\mathcal{O}})$ and the angular rates about the body fixed frame $\boldsymbol{\omega}^{\mathcal{B}} = (\omega_x^{\mathcal{B}}\ \omega_y^{\mathcal{B}}\ \omega_z^{\mathcal{B}})$. We can define the state variables that model the dynamics of a quadcopter:

$$\tilde{\boldsymbol{x}} = (\boldsymbol{p}^{\mathcal{O}}\ \ \dot{\boldsymbol{p}}^{\mathcal{O}}\ \ \boldsymbol{\psi}\ \ \boldsymbol{\omega}^{\mathcal{B}})^{\top} \in \mathbb{R}^{n_{\tilde{x}}}.$$

The control inputs to the system are the thrust forces $\boldsymbol{f}_j^{\mathcal{B}} = (f_{j,x}^{\mathcal{B}}\ f_{j,y}^{\mathcal{B}}\ f_{j,z}^{\mathcal{B}})^{\top}$ generated by the rotation of the motors in the positive direction of $\boldsymbol{z}^{\mathcal{B}}$ at the location of the j-th propeller, where $j \leq 4$. For convenience in the design of the local controller in a later stage, we consider a set of auxiliary control variables, namely $\boldsymbol{f}^{\mathcal{B}} = (f_x^{\mathcal{B}}\ f_y^{\mathcal{B}}\ f_z^{\mathcal{B}})^{\top}$ and $\boldsymbol{\tau}^{\mathcal{B}} = (\tau_x^{\mathcal{B}}\ \tau_y^{\mathcal{B}}\ \tau_z^{\mathcal{B}})^{\top}$, which encode the combined effect of the rotor thrusts in terms of total forces and torques. $\boldsymbol{f}^{\mathcal{B}}$ and $\boldsymbol{\tau}^{\mathcal{B}}$ are obtained from the $\boldsymbol{f}_j^{\mathcal{B}}$s as follows:

$$\begin{aligned}
\boldsymbol{f}^{\mathcal{B}} &= \sum \boldsymbol{f}_j{}^{\mathcal{B}}, \\
\boldsymbol{\tau}^{\mathcal{B}} &= c\boldsymbol{f}^{\mathcal{B}} + \sum \boldsymbol{p}_j{}^{\mathcal{B}} \times \boldsymbol{f}_j{}^{\mathcal{B}},
\end{aligned}$$

where $\boldsymbol{p}_j^{\mathcal{B}} = (p_{j,x}^{\mathcal{B}}\ p_{j,y}^{\mathcal{B}}\ p_{j,z}^{\mathcal{B}})$ is the position of the j-th rotor in body-fixed coordinates, and c is a parameter accounting for the aerodynamic drag. As a control input vector we consider

$$\tilde{\boldsymbol{v}} = (f_z^{\mathcal{B}}\ \ \tau_x^{\mathcal{B}}\ \ \tau_y^{\mathcal{B}}\ \ \tau_z^{\mathcal{B}})^{\top} \in \mathbb{R}^{n_{\tilde{v}}}.$$

The non-linear EoMs are easily obtained applying the Newton-Euler method. We distinguish between linear and rotational dynamics for clarity and use the less cumbersome notation in terms of state variables $\boldsymbol{x}$ and control inputs $\boldsymbol{v}$. The EoMs for translation write as:

$$\begin{aligned}
\dot{\tilde{x}}_1 &= \tilde{x}_4 \\
\dot{\tilde{x}}_2 &= x_5 \\
\dot{\tilde{x}}_3 &= x_6 \\
\dot{\tilde{x}}_4 &= \frac{\tilde{v}_1}{m}\big(\cos(\tilde{x}_9)\sin(\tilde{x}_8)\cos(\tilde{x}_7) + \sin(\tilde{x}_9)\sin(\tilde{x}_7)\big) \\
\dot{\tilde{x}}_5 &= \frac{\tilde{v}_1}{m}\big(\sin(\tilde{x}_9)\sin(\tilde{x}_8)\cos(\tilde{x}_7) - \cos(\tilde{x}_9)\sin(\tilde{x}_7)\big) \\
\dot{\tilde{x}}_6 &= \frac{\tilde{v}_1}{m}\big(\cos(\tilde{x}_8)\cos(\tilde{x}_7)\big) - g
\end{aligned} \tag{2.1}$$

where g is the acceleration due to gravity. Note that translation depends only on the total thrust $\boldsymbol{f}^{\mathcal{B}}$. Rotational dynamics of $\mathcal{B}$ with respect to $\mathcal{O}$ is written as:

$$\begin{aligned}
\dot{\tilde{x}}_7 &= \tilde{x}_{10} + \tilde{x}_{11}\sin(\tilde{x}_7)\tan(\tilde{x}_8) + \tilde{x}_{12}\cos(\tilde{x}_7)\tan(\tilde{x}_8) \\
\dot{\tilde{x}}_8 &= \tilde{x}_{11}\cos(\tilde{x}_7) - \tilde{x}_{12}\sin(\tilde{x}_7) \\
\dot{\tilde{x}}_9 &= \tilde{x}_{11}\frac{\sin\tilde{x}_7}{\cos\tilde{x}_8} + \tilde{x}_{12}\frac{\cos\tilde{x}_7}{\cos\tilde{x}_8} \\
\dot{\tilde{x}}_{10} &= J_{xx}^{-1}\big((J_{yy} - J_{zz})\tilde{x}_{11}\tilde{x}_{12} + \tilde{v}_2\big) \\
\dot{\tilde{x}}_{11} &= J_{yy}^{-1}\big((J_{zz} - J_{xx})\tilde{x}_{10}\tilde{x}_{12} + \tilde{v}_3\big) \\
\dot{\tilde{x}}_{12} &= J_{zz}^{-1}\big((J_{xx} - J_{yy})\tilde{x}_{10}\tilde{x}_{11} + \tilde{v}_4\big)
\end{aligned} \tag{2.2}$$

with J_{xx}, J_{yy}, J_{zz} being the entries of the diagonal inertia matrix J. The reader is invited to notice that the equilibrium inputs are the same at all positions $\boldsymbol{p}$ and yaw angles α [32].

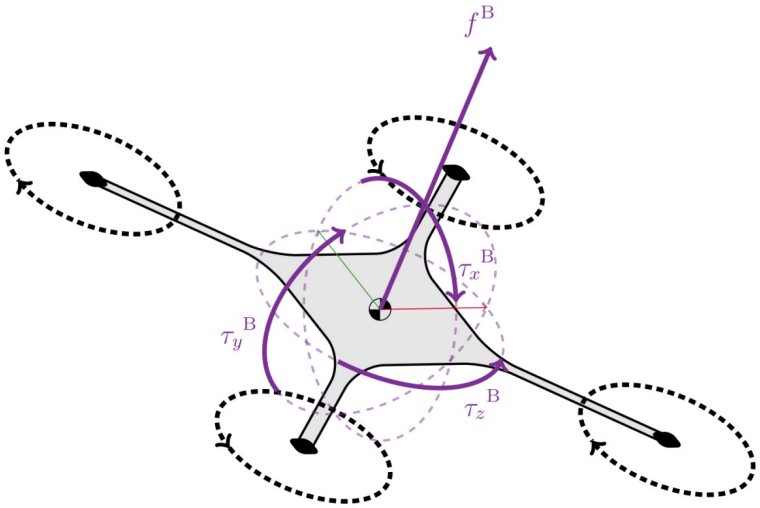


Figure 2.1: Visualization of the entries of $\tilde{\boldsymbol{v}}$.

2.2 Crazyflie 2.0: Control Architecture

As most off-the-shelf quadcopters, the Crazyflies mount reliable on-board controllers implemented by the manufacturer which allow the user to neglect the details on the propeller thrusts and directly specify a desired reference signal. Such controller, which we refer to as the *inner controller*, is fed with measurements of the angular rates $\boldsymbol{\omega}$ by an inertial measurement unit (IMU) and tracks the signal $\overline{\boldsymbol{\omega}}$ computed by a user-defined *outer controller*. Then the *inner controller* outputs the torque adjustments $\delta\boldsymbol{\tau}$ to each propeller. Regulation to an arbitrary hovering configuration $\overline{\boldsymbol{x}}$ is carried out by the *outer controller* using position and and orientation measurements to select the appropriate propeller thrusts δf_z and the already mentioned reference angular rates. As in [28, 32] we keep the *inner controller* as implemented by the manufacturer and make the following assumption:

Assumption 4 (Time-Scale Separation)**.** *The inner controller stabilizes the angular rate dynamics in (2.2). In addition, $\overline{\boldsymbol{\omega}}$ is tracked infinitely faster than $\overline{\boldsymbol{x}}$.*

This statement is motivated by the existence of measurement streams having different frequencies, namely, measurement data of $\boldsymbol{p}$ and $\boldsymbol{\psi}$ entering the *outer controller* are available at a much lower frequency than $\boldsymbol{\omega}$. We choose a cascaded configuration for the controller synthesis, as a way to independently design the *inner controller* and the *outer controller* [33]. If Assumption 4 is valid, claims on the stability of the interconnection in Fig. 2.2 can be made without any knowledge of the on-board controller. The linearized EoMs of the outer loop and the inner loop write as:

$$\begin{pmatrix} \delta\dot{\boldsymbol{p}} \\ \delta\ddot{\boldsymbol{p}} \\ \delta\dot{\boldsymbol{\psi}} \end{pmatrix} = \left.\begin{pmatrix} \frac{\partial}{\partial \boldsymbol{p}}\dot{\boldsymbol{p}} & \frac{\partial}{\partial \dot{\boldsymbol{p}}}\dot{\boldsymbol{p}} & \frac{\partial}{\partial \boldsymbol{\psi}}\dot{\boldsymbol{p}} \\ \frac{\partial}{\partial \boldsymbol{p}}\ddot{\boldsymbol{p}} & \frac{\partial}{\partial \dot{\boldsymbol{p}}}\ddot{\boldsymbol{p}} & \frac{\partial}{\partial \boldsymbol{\psi}}\ddot{\boldsymbol{p}} \\ \frac{\partial}{\partial \boldsymbol{p}}\dot{\boldsymbol{\psi}} & \frac{\partial}{\partial \dot{\boldsymbol{p}}}\dot{\boldsymbol{\psi}} & \frac{\partial}{\partial \boldsymbol{\psi}}\dot{\boldsymbol{\psi}} \end{pmatrix}\right|_{\substack{\tilde{\boldsymbol{x}}=\overline{\tilde{\boldsymbol{x}}} \\ \tilde{\boldsymbol{v}}=\overline{\tilde{\boldsymbol{v}}}}} \begin{pmatrix} \delta\boldsymbol{p} \\ \delta\dot{\boldsymbol{p}} \\ \delta\boldsymbol{\psi} \end{pmatrix} + \left.\begin{pmatrix} \frac{\partial}{\partial f_z}\dot{\boldsymbol{p}} & \frac{\partial}{\partial \boldsymbol{\omega}}\dot{\boldsymbol{p}} \\ \frac{\partial}{\partial f_z}\ddot{\boldsymbol{p}} & \frac{\partial}{\partial \boldsymbol{\omega}}\ddot{\boldsymbol{p}} \\ \frac{\partial}{\partial f_z}\dot{\boldsymbol{\psi}} & \frac{\partial}{\partial \boldsymbol{\omega}}\dot{\boldsymbol{\psi}} \end{pmatrix}\right|_{\substack{\tilde{\boldsymbol{x}}=\overline{\tilde{\boldsymbol{x}}} \\ \tilde{\boldsymbol{v}}=\overline{\tilde{\boldsymbol{v}}}}} \begin{pmatrix} \delta f_z \\ \delta\boldsymbol{\omega} \end{pmatrix}, \tag{2.3}$$

$$\delta\dot{\boldsymbol{\omega}} = \left.\frac{\partial}{\partial \boldsymbol{\omega}}\dot{\boldsymbol{\omega}}\right|_{\substack{\tilde{\boldsymbol{x}}=\overline{\tilde{\boldsymbol{x}}} \\ \tilde{\boldsymbol{v}}=\overline{\tilde{\boldsymbol{v}}}}} \delta\dot{\boldsymbol{\omega}} + J^{-1}\delta\boldsymbol{\tau}. \tag{2.4}$$

where the notation is consistent with that used in Appendix A.1. We define the actual state $\boldsymbol{x} = (\boldsymbol{p}\ \dot{\boldsymbol{p}}\ \boldsymbol{\psi})^\top \in \mathbb{R}^{n_x}$ and control input $\boldsymbol{v} = (f_z\ \boldsymbol{\omega})^\top \in \mathbb{R}^{n_v}$ and introduce the state and output

equations that we will use to design the local control laws:

$$\delta\dot{\boldsymbol{x}}(t) = A\delta\boldsymbol{x}(t) + B\delta\boldsymbol{v}(t), \tag{2.5a}$$

$$\delta\boldsymbol{y}(t) = C\delta\boldsymbol{x}(t). \tag{2.5b}$$

where $C \in \mathbb{R}^{n_y \times n_x}$ is the matrix that yields $\boldsymbol{y} = (x_1\ x_2\ x_3)^\top$.

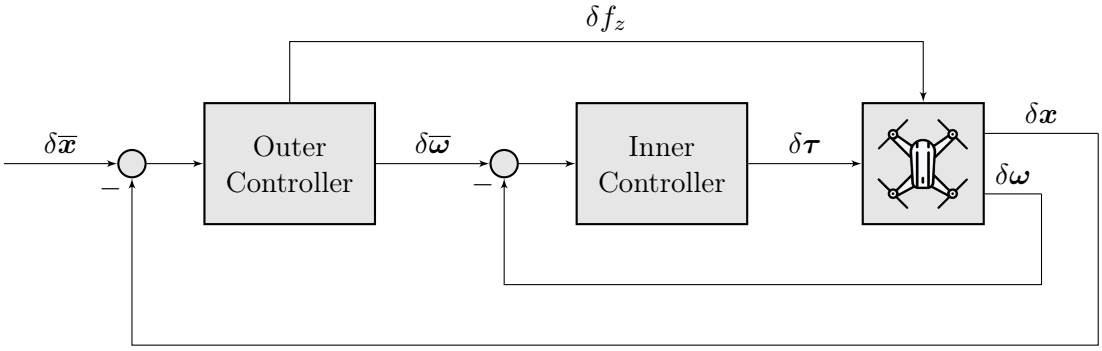


Figure 2.2: Cascade interconnection adopted in the experiments. The focus is put only on the *outer controller* as the *inner controller* is already implemented and cannot be bypassed [32].

With this in mind, the open-loop state equations are trivially obtained by putting together the linear and angular reduced EoMs of the N individual quadcopters. To keep notation consistent with the one introduced in Subsection 1.4.2, in the rest of this paper we will denote the state of the network as $\boldsymbol{x} = [\boldsymbol{x}_i]_{i\in\mathcal{I}}$, and the output vector as $\boldsymbol{y} = [\boldsymbol{y}_i]_{i\in\mathcal{I}}$, with $N_x = Nn_x$ and $N_y = Nn_y$.

2.3 FES: Stabilize then Optimize

Chapter 1 presented *formation* and *coverage* control as two relevant interpretations of the general problem of coordinating UAV-aided wireless networks. In addition, the limited sensing capabilities and computational resources of the UAVs justified a distributed control approach.
Aware of this, we want to design fully-distributed output feedback controllers to drive and maintain a UAV swarm near optimal operating conditions that are relevant in *formation* and *coverage* control. We will approach this problem using the *stabilize-then-optimize* strategy proposed in [1]. The resulting control architecture is illustrated in Fig. 2.3 and features a continuous-time physical system, namely the UAV swarm to be controlled, in closed-loop with a discrete-time ES algorithm, which serves as a trajectory planner. The latter is an update rule that computes the position setpoints $\boldsymbol{u}_i$ of each $i \in \mathcal{I}$ using local information, i.e., online position measurements $\boldsymbol{y}_i$ and $\boldsymbol{y}_{\mathcal{N}_i}$, where $\mathcal{N}_i \in \mathcal{I}$ is the set of neighbours of i in $\mathcal{G}_\mathcal{I}$.
The next chapters focus on ensuring that the obtained sampled-data closed-loop interconnection is locally input-to-state stable with respect to fluctuations in the Internet demand. In order to do that, we pre-stabilize the physical system and design appropriate trajectory planners to solve the optimal *formation* and *coverage* control objectives. When designing the planner we take inspiration from popular algorithms found in literature.
In Chapter 3, local control policies are considered that pre-stabilize the physical system (1.1). Local exponential stability of a swarm of LQR- and MPC-controlled non-linear quadcopters is formally proved in Appendix A.2 for one UAV and then extended to the multi-agent system by leveraging its decoupled dynamics. In Chapter 4, we model the *optimal formation* and *optimal coverage* objectives using generalized equations (1.2) and discuss two popular solution algorithms.

Strong regularity of the GEs and robust convergence of the proposed control algorithms are also proved in this chapter. In the rest of the paper we will refer to the trajectory planner as the FES controller.

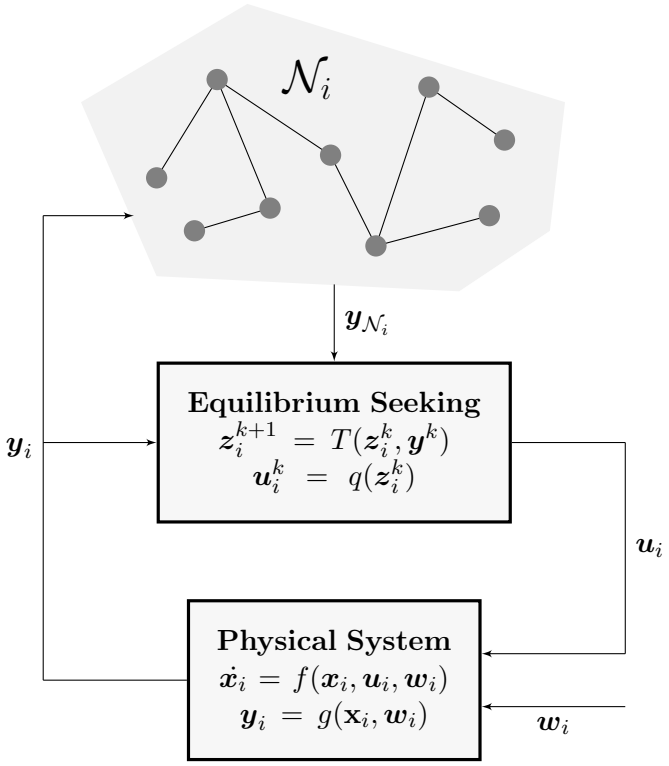


Figure 2.3: Measurements from a continuous-time dynamical multi-agent system are fed to an equilibrium seeking algorithm working in discrete time. The result is a sampled-data interconnection.

Chapter 3

Local Control Problem

The control architecture of the employed real-world quadcopters, as well as a simpler controller synthesis and parameter tuning, motivate the use of linearized EoMs. Guarantees on the local stability of the non-linear quadcopter dynamics are obtained through a stability analysis of (2.5), under two optimal state-feedback policies, namely LQR and MPC, in a disturbance-free setting. Formally, we ensure that there exist well-defined solution trajectories and steady-state mappings.

3.1 Local Stability Analysis

The elements of $\mathcal{I}$ are identical and with decoupled dynamics: if there exists a local control policy that stabilizes the individual agents, then the the whole network dynamics (1.1) can be pre-stabilized. With this in mind, we consider only one agent and omit the subscript i for convenience. Recall that the control input to the pre-stabilized UAV is a desired hovering position in $\mathcal{U} = \mathcal{FR}$, i.e., $\boldsymbol{u} = \boldsymbol{p}^\top$. The steady-state map and the steady-state input-output map are well defined

$$p(\boldsymbol{u}) = C^\top \boldsymbol{u}, \tag{3.1a}$$

$$h(\boldsymbol{u}) = g(p(\boldsymbol{u})) = \boldsymbol{u}, \tag{3.1b}$$

with C the same as in (2.5b), and $CC^\top = I_{n_u}$.
The following preparatory lemmas provide sufficient conditions for asymptotic stability of an autonomous non-linear system and will be extensively used throughout our discussion:

Lemma 3 (Lyapunov Stability Criteria [34])**.** *Consider an equilibrium point $\overline{\boldsymbol{x}} \in \mathbb{R}^{n_x}$ of the autonomous system $\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t))$. Assume that there exists a continuously differentiable function $V \colon \mathbb{R}^{n_x} \to \mathbb{R}$ and $\varepsilon_1, \varepsilon_2 > 0$ such that:*

(i) $V(\overline{\boldsymbol{x}}) = 0$ and $V(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \neq \overline{\boldsymbol{x}}$ such that $\|\boldsymbol{x} - \overline{\boldsymbol{x}}\| < \varepsilon_1$,

(ii) $\dot{V}(\boldsymbol{x}(t)) < 0$ for all $\boldsymbol{x} \neq \overline{\boldsymbol{x}}$ such that $\|\boldsymbol{x} - \overline{\boldsymbol{x}}\| < \varepsilon_2$.

Then $\overline{\boldsymbol{x}}$ is a locally asymptotically stable equilibrium. If these conditions are met for all $\boldsymbol{x} \in \mathbb{R}^{n_x}$ and V is proper, then $\overline{\boldsymbol{x}}$ is globally asymptotically stable.

Lemma 4 (Indirect Lyapunov Criterion)**.** *Consider an equilibrium point $\overline{\boldsymbol{x}} \in \mathbb{R}^{n_x}$ of the autonomous, non-linear system $\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t))$. Let $\dot{\boldsymbol{x}}(t) = A(\boldsymbol{x}(t) - \overline{\boldsymbol{x}})$ be the system obtained linearizing about $\overline{\boldsymbol{x}}$. The following statements hold:*

(i) $\overline{\boldsymbol{x}}$ is locally exponentially stable if A is Hurwitz,

(ii) $\overline{\boldsymbol{x}}$ is unstable if there exists one eigenvalue of A with strictly positive real part.

3.1.1 Infinite Horizon LQR

Let $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}, \boldsymbol{u}(t) \in \mathbb{R}^{n_u}, \boldsymbol{y}(t) \in \mathbb{R}^{n_y}$ and consider the autonomous state-output system

$$\dot{\boldsymbol{x}}(t) = (A - BK)(\boldsymbol{x}(t) - p(\boldsymbol{u}(t))), \tag{3.2a}$$

$$\boldsymbol{y}(t) = C\boldsymbol{x}(t), \tag{3.2b}$$

where (2.3) is in closed-loop with the optimal control policy of the continuous-time unconstrained Linear Quadratic Regulator (LQR). The cost matrices are $Q \in \mathbb{S}_{+}^{n_x}$, $R \in \mathbb{S}_{++}^{n_v}$ and the pair $\left(Q^{1/2}, A\right)$ is observable. Notably, the state-feedback policy is the linear function represented by the matrix

$$K = R^{-1}B^{\top}P, \tag{3.3}$$

where P solves the continuous-time Riccati Equation

$$Q + A^{\top}P + PA - PBR^{-1}B^{\top}P = 0. \tag{3.4}$$

The next lemma comes from known results and is proved in Appendix A.2:

Lemma 5. *Under Assumption 4, the cascade interconnection of (3.3) with the on-board controller stabilizes the non-linear EoMs (2.1) and (2.2), i.e., the closed-loop satisfies Assumption 1.*

3.1.2 Nominal Model Predictive Control

Consider the discrete-time, LTI open-loop system

$$\delta\boldsymbol{x}_{k+1} = \hat{A}\delta\boldsymbol{x}_k + \hat{B}\delta\boldsymbol{v}_k, \tag{3.5a}$$

$$\delta\boldsymbol{y}_k = \hat{C}\delta\boldsymbol{x}_k, \tag{3.5b}$$

where $\delta\boldsymbol{x}_k \in \mathbb{R}^{n_x}$, $\delta\boldsymbol{v}_k \in \mathbb{R}^{n_v}$, $\hat{A} = I + \tau A$, $\hat{B} = \tau B$, $\hat{C} = C$ and $\tau \in \mathbb{R}_{++}$ is the sampling period. To account for constraints in the state and the inputs, we formulate the general constrained, finite time optimal control (CFTOC) problem

$$\begin{aligned}
\underset{\delta\mathfrak{X},\,\delta\mathfrak{V}}{\text{minimize}} \quad & l_J(\delta\boldsymbol{x}_J) + \sum_{j=0}^{J-1} l(\delta\boldsymbol{x}_j, \delta\boldsymbol{v}_j) \\
\text{subject to} \quad & \delta\boldsymbol{x}_{j+1} = \hat{A}\delta\boldsymbol{x}_j + \hat{B}\delta\boldsymbol{v}_j, && \forall j \in \mathcal{J}, \\
& \delta\boldsymbol{x}_j \in \delta\mathcal{X}, && \forall j \in \mathcal{J}, \\
& \delta\boldsymbol{v}_j \in \delta\mathcal{V}, && \forall j \in \mathcal{J}, \\
& \delta\boldsymbol{x}_J \in \delta\mathcal{X}_J, \\
& \delta\boldsymbol{x}_0 = \delta\boldsymbol{x}_k,
\end{aligned} \tag{3.6}$$

where $\mathcal{J} \coloneqq \{0, \ldots, J-1\}$ is the time horizon, $\delta\mathfrak{X} = (\boldsymbol{x}_0^{\top} \ \ldots \ \boldsymbol{x}_J^{\top})^{\top}$ and $\delta\mathfrak{V} = (\boldsymbol{v}_0^{\top} \ \ldots \ \boldsymbol{v}_{J-1}^{\top})^{\top}$ are stacked vectors of decision variables, $l_J(\cdot) = \|\cdot\|_P^2$ and $l(\cdot, \odot) = \|\cdot\|_Q^2 + \|\odot\|_R^2$ are the quadratic terminal cost and quadratic stage cost, $P \in \mathbb{R}^{n_x \times n_x}$, $Q \in \mathbb{R}^{n_x \times n_x}$, $R \in \mathbb{R}^{n_v \times n_v}$ are matrices of weights, $\delta\boldsymbol{x}_k$ is the initial state. Let $\mathcal{X} \subseteq \mathbb{R}^{n_x}, \mathcal{V} \subseteq \mathbb{R}^{n_v}, \mathcal{X}_f \subseteq \mathcal{X}$ be convex sets expressing the constraints on the state, the control input and the terminal state for the non-linear EoMs, respectively: we consider the following set of constraints for the proposed linear MPC:

$$\begin{aligned}
\delta\mathcal{X} &= \{\delta\boldsymbol{x} \in \mathbb{R}^{n_x} \mid \delta\boldsymbol{x} + \overline{\boldsymbol{x}} \in \mathcal{X}\}, \\
\delta\mathcal{V} &= \{\delta\boldsymbol{v} \in \mathbb{R}^{n_v} \mid \delta\boldsymbol{v} + \overline{\boldsymbol{v}} \in \mathcal{V}\}, \\
\delta\mathcal{X}_J &= \{\delta\boldsymbol{x}_J \in \mathbb{R}^{n_x} \mid \delta\boldsymbol{x}_J + \overline{\boldsymbol{x}} \in \mathcal{X}_J\}.
\end{aligned}$$

The solution of (3.6) is a stabilizing policy for (3.5) under reasonable assumptions on the problem parameters [35]. Local exponential stability of the closed-loop is proved in [36, §5.6.3]. Below, we derive a suitable linear map that ensures satisfaction of the following specifications.

Assumption 5 (Stabilizing Feedback through MPC)**.** *The following statements hold:*

(i) $\big(A, B\big)$ *is stabilizable,*

(ii) $\mathcal{X}$ *and* $\mathcal{V}$ *contain the origin in their interior,*

(iii) the weight matrices are such that $Q \in \mathbb{S}^{n_x}_{++}, R \in \mathbb{R}^{n_u}_{++}$,

(iv) there exists a control law $K_J \in \mathbb{R}^{n_v \times n_x}$ *under which the* $\delta\mathcal{X}_J \subseteq \delta\mathcal{X}$ *is positively invariant,* $K_J(\delta\boldsymbol{x}) \in \delta\mathcal{V}$*, and* $P \in \mathbb{S}^{n_x}_{++}$ *is such that*

$$l_J(\boldsymbol{x}_{k+1}) - l_J(\boldsymbol{x}_k) \leq -l(\boldsymbol{x}_k, K_J(\boldsymbol{x}_k)).$$

Proof.
Take (i), (iii) in Assumption 5 for granted, then (iv) is satisfied under the optimal policy for an unconstrained infinite horizon LQR in discrete time

$$K_J = -(\hat{B}^\top P \hat{B} + R)^{-1} \hat{B}^\top P \hat{A},$$

where $P \in \mathbb{S}^{n_x}_{++}$ solves the discrete-time ARE. Let now $l_f(\cdot) = \| \cdot \|^2_P$ and $\delta\mathcal{X}_f$ be the largest sub-level set of l_f satisfying the state and input constraints, then $\delta\mathcal{X}_f$ is invariant by design and the terminal cost satisfies

$$\begin{aligned} l_J(\delta\boldsymbol{x}_{k+1}) - l_J(\delta\boldsymbol{x}_k) &= \delta\boldsymbol{x}_k^\top ((\hat{A} + \hat{B}K_J)^\top P (\hat{A} + \hat{B}K_J) - P)\delta\boldsymbol{x}_k \\ &\leq \delta\boldsymbol{x}_k^\top (\hat{A}^\top P \hat{A} + K_J^\top \hat{B}^\top P \hat{A} - K_J^\top R K_J - P)\delta\boldsymbol{x}_k \\ &= -\delta\boldsymbol{x}_k^\top (Q - K_J^\top R K_J)\delta\boldsymbol{x}_k, \end{aligned}$$

hence verifying the statement. ■

Lemma 6 (MPC Feedback Policy)**.** *Consider the mp-QP formulation of (3.6) as in [37]. Let its cost function be positive definite and* $\delta\mathcal{X}$ *be a convex set. Then the set of feasible initial conditions* $\delta\mathcal{X}_0 \subseteq \delta\mathcal{X}$ *is convex, the optimizer* $\delta\mathfrak{V}^*(\delta\boldsymbol{x})\colon \delta\mathcal{X}_0 \to \mathbb{R}^{Tn_v}$ *is continuous and piecewise affine, and the optimal cost is convex, continuous and piecewise quadratic.*

Thus, manipulation of a CFTOC problem into a multi-parametric quadratic program (mp-QP) gives information on the shape of the optimal control policy [37], [38], i.e.,

$$\delta\boldsymbol{v}^* = \kappa^*(\delta\boldsymbol{x}) = \Xi_0\, \delta\mathfrak{V}^*(\delta\boldsymbol{x}), \tag{3.7}$$

where $\delta\mathfrak{V}^*$ yields the optimizing control sequence defined on the set of admissible initial conditions $\delta\mathcal{X}_0$ and $\Xi_0 \in \mathbb{R}^{n_v \times Tn_v}$ is a matrix selecting the control input to apply at time zero. In our discussion we use the piecewise affine formulation of $\kappa^*(\delta\boldsymbol{x}) = K_j\delta\boldsymbol{x} + h_j$ for all $\delta\boldsymbol{x} \in \mathcal{CR}_j$, where each $\mathcal{CR}_j$ is a convex polyhedral region and $\delta\mathcal{X}_0 = \bigcup_j \mathcal{CR}_j$. Given $\delta\boldsymbol{x}_k \in \mathcal{CR}_j$ the resulting closed-loop system is

$$\boldsymbol{x}_{k+1} = (\hat{A} + \hat{B}K_j)\boldsymbol{x}_k - (\hat{A} + \hat{B}K_j - I)p(\boldsymbol{u}_k) + h_j, \tag{3.8a}$$

$$\boldsymbol{y}_k = \hat{C}\boldsymbol{x}_k. \tag{3.8b}$$

Lemma 7. *Under Assumption 4, the cascade interconnection of (3.7) with the on-board controller stabilizes the non-linear EoMs (2.1) and (2.2), i.e., the closed-loop satisfies Assumption 1.*

Again, we refer the reader to Appendix A.2 for a formal proof of this statement.

Chapter 4

Optimality Condition and Algorithms

In this chapter, we give a rigorous formulation for autonomous *formation control* [3, 18] and optimal *geographic coverage* [6, 8], and then present popular solution strategies [25, 39].

4.1 Autonomous Online Distributed Formation Control

We have already addressed the limitations arising in a multi-agent scenario, e.g, the finite sensing radius and computational resources that allow one agent to process only local information. However, additional restrictions on power consumption for the single agent step in, as the motion and communication effort seriously affect the battery life-time. This is especially true for quadcopters, which are only suitable for short-lasting missions. Design of a global control objective that encompasses all of them is hard, as self-interested agents are embedded in a network that is supposed to behave cohesively [3]. With this in mind, in the next section we propose an energy-aware formation control technique for quadcopters (*followers*) to enable reliable wireless communication paths between selfish agents (*masters*) that are geographically far apart, and restrict ourselves to LTI graphs. Employing a distributed, online FES control framework, we indirectly model savings in the energy allocated for motion and inter-agent communication.

4.1.1 GE Formulation

Let $\mathcal{I}$ be the set of *followers*, $\mathcal{M} = \{N+1, \dots, N+M\}$ the set of *masters*, $\mathcal{G}(\mathcal{I} \cup \mathcal{M}, \mathcal{E})$ a linear time-invariant graph, and $\mathcal{G}_{\mathcal{I}}(\mathcal{I}, \mathcal{E}_{\mathcal{I}})$ and $\mathcal{G}_{\mathcal{M}}(\mathcal{M}, \mathcal{E}_{\mathcal{M}})$ be the two subgraphs of $\mathcal{G}$ induced by $\mathcal{I}$ and $\mathcal{M}$ respectively. To simplify notation we define $\mathcal{E}_{\mathcal{M}}^{\mathcal{I}} = \mathcal{E} \setminus (\mathcal{E}_{\mathcal{I}} \cup \mathcal{E}_{\mathcal{M}})$ to be the set of edges $\{i, j\} \in \mathcal{E}$ linking *followers* to *masters*. We then model the *masters* as exogenous disturbances $\boldsymbol{w} \in \mathcal{W}$, where $\mathcal{W} = \bigtimes_{j \in \mathcal{M}} \mathcal{W}_j$ and $\mathcal{W}_j = \mathcal{FR}$, for all $j \in \mathcal{M}$. In particular, we allow each $j \in \mathcal{M}$ to freely move inside the $\mathcal{FR}$ and thus influence the shape of $\mathcal{G}$.
We want to design an optimization problem whose solution $\boldsymbol{u}^* \in \mathcal{U}$ is a spatial configuration of $\mathcal{I}$ that maximizes the quality of wireless communication between a pair of agents of $\mathcal{M}$ [11, 22]. This is the same as minimizing the energy spent for inter-agent communication [12, 23]. By modelling this process as a function of the squared Euclidean distance between any pair of nodes in the graph, we get the constrained QP:

$$\min_{\boldsymbol{u}} \quad \mathbb{J}(\boldsymbol{u}, \boldsymbol{w}) + \iota_{\mathcal{U}}(\boldsymbol{u}), \tag{4.1}$$

where the following global cost is used

$$\mathbb{J}(\boldsymbol{u}, \boldsymbol{w}) = \frac{1}{2} \left(\sum_{\{i,j\} \in \mathcal{E}_{\mathcal{I}}} \|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2 + \sum_{\{i,j\} \in \mathcal{E}_{\mathcal{M}}^{\mathcal{I}}} \|\boldsymbol{u}_i - \boldsymbol{w}_j\|^2 \right). \tag{4.2}$$

Before moving on, let us make some considerations on (4.2) to spot undesired network behaviours. First, we identify $M \geq 2$ as a necessary condition for it to be well-posed: multiple *masters* will indeed stretch the network configuration, i.e., ensure that the *followers* do not converge to the same location. As an heuristic measure for collision avoidance, we make the following assumption.

Assumption 6 (Structure of $\mathcal{G}_\mathcal{I}$ and $\mathcal{G}$)**.** *The following statements on the structure of the interaction graphs hold:*

(i) $\mathcal{G}_\mathcal{I}$ *and* $\mathcal{G}$ *are both connected,*

(ii) For all $i \in \mathcal{I}$ *there exists one path connecting any pair of* $m_1, m_2 \in \mathcal{M}$ *that includes* i*.*

This rules out the scenario in which two or more *followers* have the same minimizer in $\mathbb{R}^3$. Note that this assumption does aim at being a general solution for ensuring collision avoidance, but rather a first and somewhat comprehensive formulation that can be readily implemented in our experiments. It is trivial to prove equivalence of (4.2) and the following interdependent optimization problems, i.e., non cooperative game,

$$\forall i \in \mathcal{I} \ : \ \min_{\boldsymbol{u}_i} \quad \mathbb{J}_i(\boldsymbol{u}_i, \boldsymbol{u}_{-i}, \boldsymbol{w}) + \iota_{\mathcal{U}_i}(\boldsymbol{u}_i), \tag{4.3}$$

where each decision maker in $\mathcal{I}$ is assigned the quadratic local utility

$$\mathbb{J}_i(\boldsymbol{u}_i, \boldsymbol{u}_{-i}, \boldsymbol{w}) = \frac{1}{2}\left(\sum_{j \in \mathcal{I} \cap \mathcal{N}_i} \|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2 + \sum_{j \in \mathcal{M} \cap \mathcal{N}_i} \|\boldsymbol{u}_i - \boldsymbol{w}_j\|^2 \right). \tag{4.4}$$

Let us now define the pseudo-gradient map $\mathbb{F}(\boldsymbol{u}, \boldsymbol{w}) = F_i(\boldsymbol{u}, h(\boldsymbol{u}, \boldsymbol{w})) = [F_i(\boldsymbol{u}, h(\boldsymbol{u}, \boldsymbol{w}))]_{i \in \mathcal{I}}$ obtained by stacking the partial gradients $F_i(\boldsymbol{u}, h(\boldsymbol{u}, \boldsymbol{w})) \coloneqq \nabla_{\boldsymbol{u}_i} \mathbb{J}_i(\boldsymbol{u}_i, \boldsymbol{u}_{-i}, \boldsymbol{w})$ of the local utilities. Each $\mathbb{J}_i$ is continuously differentiable and strongly convex in its decision variable $\boldsymbol{u}_i$ and hence the unique optimal formation is obtained by solving the system

$$0 \in \nabla_{\boldsymbol{u}_i} \mathbb{J}_i(\boldsymbol{u}_i, \boldsymbol{u}_{-i}, \boldsymbol{w}) + \mathcal{N}_{\mathcal{U}_i}(\boldsymbol{u}_i), \quad \forall i \in \mathcal{I},$$

where each partial gradient evaluates to the affine function of $\boldsymbol{u}$

$$F_i(\boldsymbol{u}, h(\boldsymbol{u}, \boldsymbol{w})) = (L_i^{\mathcal{I}\mathcal{I}} \otimes I_{n_u})\boldsymbol{u} + (L_i^{\mathcal{M}\mathcal{I}} \otimes I_{n_u})\boldsymbol{w}, \quad \forall i \in \mathcal{I}, \tag{4.5}$$

where $L \in \mathbb{R}^{(N+M)\times(N+M)}$ denotes the binary Laplacian matrix of $\mathcal{G}$, and L_i its i-th row. In particular, $L_i^{\mathcal{I}\mathcal{I}} \in \mathbb{R}^{1\times N}$ and $L_i^{\mathcal{M}\mathcal{I}} \in \mathbb{R}^{1\times M}$ serve to distinguish the edges in $\mathcal{E}_\mathcal{I}$ from those in $\mathcal{E}_\mathcal{M}^\mathcal{I}$. We obtain the following parametric GE:

$$0 \in \underbrace{(L^{\mathcal{I}\mathcal{I}} \otimes I_{n_u})\boldsymbol{z} + (L^{\mathcal{M}\mathcal{I}} \otimes I_{n_u})\boldsymbol{w}}_{\mathbb{G}(\boldsymbol{z},\boldsymbol{w})} + \underbrace{\mathcal{N}_\mathcal{U}(\boldsymbol{z})}_{\mathcal{A}(\boldsymbol{z})}, \tag{4.6}$$

where $\boldsymbol{u} = q(\boldsymbol{z}) = \boldsymbol{z}$ is an auxiliary variable used to express the reference deployment of the pre-stabilized system, $\mathcal{N}_\mathcal{U}$ is the sub-differential of $\iota_\mathcal{U}$.

The following statement ensures that (4.6) admits finitely many isolated trajectories, i.e., that the designed *formation control* problem is well posed:

Lemma 8. *The generalized equation (4.6) satisfies Assumption 2.*

Proof.

(i) Let $\boldsymbol{z}', \boldsymbol{z}'' \in \mathbb{R}^{N_u}$. The map q is globally 1-Lipschitz

$$\|q(\boldsymbol{z}'') - q(\boldsymbol{z}')\| = \|\boldsymbol{z}'' - \boldsymbol{z}'\|.$$

(ii) $G(\boldsymbol{z}, \boldsymbol{s})$ is an affine function between finite dimensional spaces and therefore continuously differentiable with respect to $\boldsymbol{z}$.

(iii) To prove strong regularity, we note that $\mathbb{J}(\boldsymbol{u}, \boldsymbol{w})$ is obtained as the summation of positive definite quadratic functions. Therefore it is a positive definite quadratic function, i.e., $(L^{\mathcal{I}\mathcal{I}} \otimes I_{n_u}) \in \mathbb{S}^{N_u}_{++}$ and is invertible. Strong regularity of the GE follows from this fact.

Assumption 2 is verified. ■

4.1.2 Projected Pseudo-Gradient Descent

To solve the non-cooperative game (4.4), we use projected pseudo-gradient descent (PPGD) [40, 41, 42], which is summarized in Algorithm 1. At each time step $k \in \mathbb{Z}_+$, agent $i \in \mathcal{I}$ is provided with the output measurements of its neighbours $\boldsymbol{y}^k_{\mathcal{N}_i}$, as well as its own $\boldsymbol{y}^k_i$. This information is used to take a step in the negative direction of the partial gradient (4.5), i.e.,

$$\hat{\boldsymbol{z}}^k_i = \Big(\boldsymbol{z}_i + \mu \nabla_{\boldsymbol{z}_i} \mathbb{J}_i(\boldsymbol{z}_i, \boldsymbol{z}_{-i}, \boldsymbol{w})\Big)\Big|_{\substack{\boldsymbol{z}=\boldsymbol{y} \\ \boldsymbol{w}=\boldsymbol{w}^k}},$$

where $\mu \in \mathbb{R}_{++}$ is a fixed step-size. The obtained result is projected onto the intersection of the action set $\mathcal{U}_i$ and the closed ball $\mathcal{Q}(\boldsymbol{y}^k_i, \nu)$ of radius $\nu \in \mathbb{R}_{++}$, centered in $\boldsymbol{y}^k_i \in \mathcal{U}_i$.

Lemma 9. *Algorithm 1 verifies Assumption 3 [24, 43].*

Proof.

(i) $\Longleftarrow$ **:** Assume that $\boldsymbol{z}$ solves (4.6), i.e. $\boldsymbol{z} = -(L_{11} \otimes I_{n_u})^{-1}(L^{\mathcal{M}\mathcal{I}} \otimes I_{n_u})\boldsymbol{w}$. For this value of $\boldsymbol{z}$ the pseudo-gradient vanishes and we get

$$\mathbb{T}(\boldsymbol{z}, \boldsymbol{w}) = \boldsymbol{z} + \mu \mathbb{G}(\boldsymbol{z}, \boldsymbol{w}) = \boldsymbol{z}.$$

$\Longrightarrow$ **:** $\mathbb{T}(\boldsymbol{z}, \boldsymbol{w}) = \boldsymbol{z}$ implies that $\nabla_z \mathbb{J}(\boldsymbol{u}, \boldsymbol{w}) = 0$. $\mathbb{G}(\boldsymbol{u}, \boldsymbol{w})$ is the same as the gradient, from equivalence of (4.1) and (4.3), meaning that $\boldsymbol{z}$ also solves the GE.

(ii) Fix $\boldsymbol{w} \in \mathcal{W}$ and let $\overline{\boldsymbol{z}} \in S(\boldsymbol{w})$ as in the previous point. The single-valued mapping $\mathbb{G}(\boldsymbol{z}, \boldsymbol{w})$ is cocoercive

$$\begin{aligned}(\mathbb{G}(\boldsymbol{z}, \boldsymbol{w}) - \mathbb{G}(\overline{\boldsymbol{z}}, \boldsymbol{w}))^\top (\boldsymbol{z} - \overline{\boldsymbol{z}}) &= (\boldsymbol{z} - \overline{\boldsymbol{z}})^\top (L^{\mathcal{I}\mathcal{I}} \otimes I_{n_u})(\boldsymbol{z} - \overline{\boldsymbol{z}}) \\ &\geq \underbrace{\lambda_{\min}(L^{\mathcal{I}\mathcal{I}} \otimes I_{n_u})}_{\alpha_4} \|\boldsymbol{z} - \overline{\boldsymbol{z}}\|^2,\end{aligned}$$

and strongly monotone

$$(\mathbb{G}(\boldsymbol{z}, \boldsymbol{w}) - \mathbb{G}(\overline{\boldsymbol{z}}, \boldsymbol{w}))^\top (\boldsymbol{z} - \overline{\boldsymbol{z}}) \leq \underbrace{\lambda_{\max}(L^{\mathcal{I}\mathcal{I}} \otimes I_{n_u})}_{\alpha_5} \|\boldsymbol{z} - \overline{\boldsymbol{z}}\|^2.$$

since $(L^{\mathcal{I}\mathcal{I}} \otimes I_{n_u}) \in \mathbb{S}^{N_u}_{++}$. By proving the following inequality

$$\begin{aligned}\mathbb{G}(\mathbb{T}(\boldsymbol{z}, \boldsymbol{w}), \boldsymbol{w}) &= (L^{\mathcal{I}\mathcal{I}} \otimes I_{n_u})\mathbb{T}(\boldsymbol{z}, \boldsymbol{w}) + (L^{\mathcal{M}\mathcal{I}} \otimes I_{n_u})\boldsymbol{w} \\ &= (L^{\mathcal{I}\mathcal{I}} \otimes I_{n_u})(\boldsymbol{z} - \mu \mathbb{G}(\boldsymbol{z}, \boldsymbol{w})) + (L^{\mathcal{M}\mathcal{I}} \otimes I_{n_u})\boldsymbol{w} \\ &= \mathbb{G}(\boldsymbol{z}, \boldsymbol{w}) - \mu (L^{\mathcal{I}\mathcal{I}} \otimes I_{n_u})\big((L^{\mathcal{I}\mathcal{I}} \otimes I_{n_u})\boldsymbol{z} + (L^{\mathcal{M}\mathcal{I}} \otimes I_{n_u})\boldsymbol{w}\big) \\ &= \mathbb{G}(\boldsymbol{z}, \boldsymbol{w}) - \mu (L^{\mathcal{I}\mathcal{I}} \otimes I_{n_u})(\mathbb{G}(\boldsymbol{z}, \boldsymbol{w}) - \mathbb{G}(\overline{\boldsymbol{z}}, \boldsymbol{w})) \\ &\leq \mathbb{G}(\boldsymbol{z}, \boldsymbol{w}) - \underbrace{\mu \alpha_4^2}_{\alpha_6} \|\boldsymbol{z} - \overline{\boldsymbol{z}}\|^2\end{aligned}$$

we show that $\mathbb{G}$ is a valid *merit function* for the dynamics of Algorithm 1.

(iii) This choice for the learning rate $\mu \in \left(0,\ 2\,\frac{\lambda_{\max}(L^{\mathcal{II}} \otimes I_{n_u})}{\lambda^2_{\min}(L^{\mathcal{II}} \otimes I_{n_u})}\right)$ satisfies the last requirement.

Projected pseudo-gradient descent satisfies Assumption 3 and is linearly convergent. ■

Algorithm 1 Projected Pseudo-Gradient Descent

Initialization: μ, ν, $\boldsymbol{z}^0 \in \mathcal{U}$
Iteration: At time k, agent i performs:
$\hat{\boldsymbol{z}}_i^k = \boldsymbol{z}_i^k - \mu \nabla_{z_i} \mathbb{J}_i(\boldsymbol{z}_i^k, \boldsymbol{z}_{-i}^k, \boldsymbol{w}^k)$
$\Omega_i^k = \mathcal{U}_i \cap \mathcal{Q}(\boldsymbol{z}_i^k, \nu)$
$\boldsymbol{z}_i^{k+1} = \Pi_{\Omega_i^k} \hat{\boldsymbol{z}}_i^k$

4.2 Autonomous Online Distributed Optimal Coverage

Consider the problem of placing resources in a geographic area to attain the most satisfaction of those exploiting them for their needs. As already mentioned, this encompasses a broad spectrum of real-world applications, from dynamic Internet services and sensor networks operating in rapidly changing environments to the more classic *locational optimization* of facilities [44]. In this Section, we focus on quadcopters' swarms for spatial *optimal coverage* of reliable Internet services in environments with (possibly moving) users. Mirroring Section 4.1, we first formalize the distributed optimal coverage problem, and then adapt a popular ES algorithm for online settings [25, 15, 16, 39].

4.2.1 GE Formulation

We base this content on the introductory material that is found in Subsection 1.4.3 and the references therein. Recall that $\boldsymbol{u}$ is the stacked vector of decision variables of the N-agent optimization problem, labelled by $i \in \mathcal{I}$ and $\mathcal{U}_i$ is a compact polytope of allowed spatial configurations for i. Leveraging Lemma 1 we express the global coverage performance using the cost functional:

$$\begin{aligned} \mathbb{J}(\boldsymbol{u}, \mathfrak{R}(\boldsymbol{u})) &= \int_{\mathcal{FR}} \left(\min_{i \in \{1,\dots,N\}} \|\boldsymbol{u}_i - \boldsymbol{q}\|^2 \right) \varrho(\boldsymbol{q}) d\boldsymbol{q} \\ &= \sum_{i \in \mathcal{I}} \int_{\mathcal{R}_i} \|\boldsymbol{u}_i - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q}, \end{aligned} \tag{4.7}$$

where $\varrho \in \mathcal{C}^2(\mathcal{FR})$ is a probability density function expressing how likely the event of a person requiring a Internet connection is on $\mathcal{FR}$, and $\mathfrak{R} = \{\mathcal{R}_1, \dots, \mathcal{R}_N\}$ the Voronoi partition of the flying region, generated by $\boldsymbol{u}$. As shown in [16], the functional (4.7) is $\mathcal{C}^2(\Gamma)$ if $\mathcal{FR}$ is convex and compact, and ϱ is $\mathcal{C}^2(\mathcal{FR})$. We decide to omit the dependence on $\boldsymbol{w}$ since this problem does not involve exogenous disturbances.

Before we present a *Voronoi-distributed* formulation for the global objective we highlight the need for an efficient communication protocol for detecting the set Voronoi neighbours of one agent, and computing its Voronoi region. Computation of the Voronoi region using the definition in (1.8) is only feasible in a centralized setting, as it would require full-knowledge on the position of all agents, as well as huge computational resources. It is assumed that each agent $i \in \mathcal{I}$ can detect every other agents within a ball $\mathcal{Q}(\boldsymbol{u}_i, r_i)$ centered at $\boldsymbol{u}_i$, where $r_i \in \mathbb{R}_{++}$ is an adaptable sensing radius. Further, we consider each agent to run Algorithm 2 at each round of communication. As

in [6], we denote as

$$\mathcal{R}(\boldsymbol{u}_i, r_i) = \mathcal{Q}(\boldsymbol{u}_i, r_i) \cap \left(\bigcap_{\boldsymbol{u}_j \in \mathcal{Q}(\boldsymbol{u}_i, r_i)} \mathcal{H}_i^j \right) \tag{4.8}$$

the best estimate for $\mathcal{R}_i$ associated to i that can be achieved using a sensing radius r_i. (4.7) can

Algorithm 2 Adjustable Sensing Radius Algorithm [6]

Iteration: At time k, agent i performs:
- Initialize the sensing radius r_i
- Detect agents within r_i
- Update $\mathcal{N}_i^k$
- Compute $\mathcal{R}_i^k := \mathcal{R}(\boldsymbol{u}_i^k, r_i)$
- **while** $r_i \leq 2 \max_{\boldsymbol{u}_i \in \mathcal{R}_i^k} \|\boldsymbol{u}_i^k - \boldsymbol{q}\|$ **do**
  - $r_i \leftarrow 2r_i$
  - Detect agents within r_i
  - Update $\mathcal{N}_i^k$
  - Update $\mathcal{R}_i^k := \mathcal{R}(\boldsymbol{z}_i^k, r_i)$
- $r_i \leftarrow 2 \max_{q \in \mathcal{R}_i^k} \|\boldsymbol{u}_i^k - \boldsymbol{q}\|$
- $\mathcal{R}_i \leftarrow \mathcal{R}_i^k$

be reformulated in terms of local utility functions of the form

$$\forall i \in \mathcal{I} \; : \; \mathbb{J}_i\big(\boldsymbol{u}_i, \boldsymbol{u}_{-i}, \mathfrak{R}(\boldsymbol{u}_i, \boldsymbol{u}_{-i})\big) = \sum_{j \in i \cup \mathcal{N}_i} \int_{\mathcal{R}_j} \|\boldsymbol{u}_j - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q}. \tag{4.9}$$

The non-unique minimizers of (4.7) are obtained as in Section 4.1 by solving

$$0 \in \nabla_{\boldsymbol{u}} \mathbb{J}(\boldsymbol{u}) + \mathcal{N}_{\mathcal{U}}(\boldsymbol{u}) \tag{4.10}$$

where the gradient is defined as

$$\nabla_{\boldsymbol{u}} \mathbb{J}(\boldsymbol{u}, \mathfrak{R}(\boldsymbol{u})) = 2 \left[\int_{\mathcal{R}_i(\boldsymbol{u})} \boldsymbol{u}_i \varrho(\boldsymbol{q}) d\boldsymbol{q} - \int_{\mathcal{R}_i(\boldsymbol{u})} \boldsymbol{q} \varrho(\boldsymbol{q}) d\boldsymbol{q} \right]_{i \in \mathcal{I}}. \tag{4.11}$$

In (4.11) we explicitly stated the dependence of the Voronoi partition on $\boldsymbol{u}$ to avoid confusion. For the complete derivation of $\mathbb{F}$ we refer the reader to Appendix B.1. With this in mind, we get the following GE:

$$0 \in \underbrace{2 \left[\boldsymbol{z}_i \int_{\mathcal{R}_i(\boldsymbol{z})} \varrho(\boldsymbol{q}) d\boldsymbol{q} - \int_{\mathcal{R}_i(\boldsymbol{z})} \boldsymbol{q} \varrho(\boldsymbol{q}) d\boldsymbol{q} \right]_{i \in \mathcal{I}}}_{\mathbb{G}(\boldsymbol{z})} + \underbrace{\mathcal{N}_{\mathcal{U}}(\boldsymbol{z})}_{\mathcal{A}(\boldsymbol{z})} \tag{4.12}$$

where $\boldsymbol{s} = h(\boldsymbol{u}, \boldsymbol{w}) = \boldsymbol{u}$ is the output at steady state, and $\boldsymbol{u} = q(\boldsymbol{z}) = \boldsymbol{z}$ is the decision variable of the FES controller. Clearly the equality holds only if $\boldsymbol{z}_i$ is simultaneously the generator and the center of mass of $\mathcal{R}_i$ for all $i \in \mathcal{I}$, as stated in Lemma 1. This result is the starting point for the majority of algorithms used to solve problems of optimal coverage as we will see in the next section. As in [6, 8, 14] we assume that (4.7) has a finite number of stationary points. Proving existence of a finite number of stationary points for multi-dimensional Voronoi coverage is an open problem in computational geometry [25, 15]. However, we conjecture that special classes of functions (e.g., strictly log-concave densities and the exponential family) defined on compact polytopes can satisfy the aforementioned property.

Proof.

(i) Let $\boldsymbol{z}', \boldsymbol{z}'' \in \mathbb{R}^{N_u}$. The map q is globally 1-Lipschitz

$$\|q(\boldsymbol{z}'') - q(\boldsymbol{z}')\| = \|\boldsymbol{z}'' - \boldsymbol{z}'\|.$$

(ii) The functional $\mathbb{J}(\boldsymbol{z}, \mathbf{R}(\boldsymbol{z}))$ is $\mathcal{C}^2(\Gamma)$, i.e. it is differentiable everywhere except on the discrete set of other agents locations [16]. This suffices to prove that $\mathbb{G}$ is continuously differentiable almost everywhere.

(iii) Let $\overline{\boldsymbol{z}} \in \mathbb{R}^{N_u}$ be a stationary point of (4.7). A closed form for the second derivative of the cost functional is obtained developing the result found in Section B.1.

Lemma 10 (Hessian Matrix at Stationary Points [25])**.** *At the stationary point $\overline{\boldsymbol{z}}$ such that $\overline{\boldsymbol{z}} = LL(\overline{\boldsymbol{z}})$, we have*

$$\nabla^2_{\boldsymbol{z}} \mathbb{J}(\boldsymbol{z}) = 2W_{\mathfrak{R}} \left(I_{N_u} - \nabla_{\boldsymbol{z}} LL(\boldsymbol{z})\right) \tag{4.13}$$

where $LL\colon \mathbb{R}^{N_u} \to \mathbb{R}^{N_u}$ is the Lloyd map (see Subsection 1.4.4) and $W_{\mathfrak{R}}\colon \mathcal{U} \to \mathbb{R}^{N_u \times N_u}$ a diagonal matrix containing the mass of each Voronoi cell.

Recall that, by assumption, only a finite number of locations satisfies the first order conditions and hence the Hessian matrix in (4.13) is non-singular. Strong regularity follows from invertibility of the truncated inverse $\mathbb{G}^{-1}$ which in turn is ensured by the following theorem.

Lemma 11 (Inverse Function Theorem [45])**.** *Let $\mathcal{J} \in \mathbb{R}^{N_u}$ be an open set and the map $\mathbb{G}\colon \mathcal{J} \to \mathbb{R}^{Nn_u}$ be $\mathcal{C}^1(\mathcal{J})$. If there exists $\boldsymbol{z} \in \mathcal{J}$ such that $\nabla_{\boldsymbol{z}} \mathbb{G}(\boldsymbol{z})$ is invertible, then there there exist also neighbourhoods $\mathcal{J}'$ of $\boldsymbol{z}$ and $\mathcal{K}'$ of $\mathbb{G}(\boldsymbol{z})$ such that $\mathbb{G}\colon \mathcal{J}' \to \mathcal{K}'$ is invertible and $\mathbb{G}^{-1}\colon \mathcal{K}' \to \mathcal{J}'$ is $\mathcal{C}^1(\mathcal{K}')$.*

Assumption 2 is met under the assumption of finitely many stationary points. Future work will focus on finding conditions on the probability density functions defined on compact polytopes for which (4.13) is non-singular. ■

4.2.2 Lloyd Iteration

In this Section, we discuss the Lloyd iteration, summarized in Algorithm 3, as a candidate discrete-time algorithm that solves the *optimal coverage* problem. In particular, the Lloyd iteration is a fixed-point iteration of the so called Lloyd map [15]. Position measurements coming from the neighbouring UAVs are used to modify the following iterative update rule:

$$\boldsymbol{s}^k = \boldsymbol{z}^k, \tag{4.14a}$$

$$T(\boldsymbol{z}^k, \boldsymbol{s}^k) = \left[\frac{\int_{\mathcal{R}_i(\boldsymbol{s}^k)} \boldsymbol{q} \varrho(\boldsymbol{q}) d\boldsymbol{q}}{\int_{\mathcal{R}_i(\boldsymbol{s}^k)} \varrho(\boldsymbol{q}) d\boldsymbol{q}} \right]_{i \in \mathcal{I}}, \tag{4.14b}$$

which we rewrite in a more compact way as in (1.3)

$$\mathbb{T}(\boldsymbol{z}^k, \boldsymbol{w}^k) = \left[\frac{\int_{\mathcal{R}_i(\boldsymbol{z}^k)} \boldsymbol{q} \varrho(\boldsymbol{q}) d\boldsymbol{q}}{\int_{\mathcal{R}_i(\boldsymbol{z}^k)} \varrho(\boldsymbol{q}) d\boldsymbol{q}} \right]_{i \in \mathcal{I}}. \tag{4.15}$$

Lemma 12 (Discrete-Time Lloyd Descent [6])**.** *Consider the continuous map $T\colon \mathbb{R}^{N_u} \to \mathbb{R}^{N_u}$ satisfying the statements:*

(i) $\forall i \in \mathcal{I}$, $\|T_i(\boldsymbol{z}) - \overline{\boldsymbol{z}}_i\| \leq \|\boldsymbol{z}_i - \overline{\boldsymbol{z}}_i\|$ *where* T_i *is the* i*-th element in (4.14b) and* $\overline{\boldsymbol{z}}_i$ *the current centroid for agent* i*,*

(ii) if $\boldsymbol{z}$ *is not a centroidal configuration, then there exists* $j \in \mathcal{I}$ *such that* $\|T_j(\boldsymbol{z}) - \overline{\boldsymbol{z}}_j\| \leq \|\boldsymbol{z}_j - \overline{\boldsymbol{z}}_j\|$.

Let $\boldsymbol{z}^0 \in \Gamma$ *denote the initial network configuration. Then, the sequence* $\{T^k(\boldsymbol{z}^0) \mid k \in \mathbb{N}_{++}\}$ *converges to the set of centroidal Voronoi configurations.*

Proof.

(i) $\Longleftarrow$ **:** Follows from Lemma 1.
$\Longrightarrow$ **:** Is obtained by substitution into (4.11), i.e.,

$$\begin{aligned} \boldsymbol{z} = \mathbb{T}(\boldsymbol{z}) \Longrightarrow \nabla_{\boldsymbol{z}} \mathbb{J}(\boldsymbol{z}, \mathfrak{R}(\boldsymbol{z})) &= W_{\mathfrak{R}}(\boldsymbol{z}))(\boldsymbol{z} - LL(\boldsymbol{z})) \\ &= W_{\mathfrak{R}}(\boldsymbol{z}))(\boldsymbol{z} - \mathbb{T}(\boldsymbol{z})) \\ &= W_{\mathfrak{R}}(\boldsymbol{z}))(\boldsymbol{z} - \boldsymbol{z}) \\ &= 0 \end{aligned}$$

which is precisely the expression of S.

(ii) Consider (4.7) as a candidate *merit function* for the Lloyd algorithm, and $\mathfrak{D}$ a partition of $\mathcal{FR}$ as described in Subsection 1.4.3. Using the Huygens-Steiner theorem [6] we rewrite the global cost as

$$\mathbb{J}(\boldsymbol{z}, \mathfrak{D}) = \sum_{i \in \mathcal{I}} \mathcal{J}_{\mathcal{D}_i, C_{\mathcal{D}_i}} + \sum_{i \in \mathcal{I}} W_{\mathcal{D}_i} \|\boldsymbol{z}_i - C_{\mathcal{D}_i}\|^2,$$

where we denote the mass of $\mathcal{D}_i$, the position of its centroid $C_{\mathcal{D}_i}$, and its moment of inertia about $C_{\mathcal{D}_i}$ respectively as:

$$W_{\mathcal{D}_i} = \int_{\mathcal{D}_i} \varrho(\boldsymbol{q}) d\boldsymbol{q}, \quad C_{\mathcal{D}_i} = W_{\mathcal{D}_i}^{-1} \int_{\mathcal{D}_i} \boldsymbol{q} \varrho(\boldsymbol{q}) d\boldsymbol{q}, \quad \mathcal{J}_{\mathcal{D}_i, C_{\mathcal{D}_i}} = \int_{\mathcal{D}_i(\boldsymbol{z})} \|\boldsymbol{z}_i - C_{\mathcal{D}_i}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q}.$$

If $\|\boldsymbol{z}_i' - C_{\mathcal{D}_i}\| \leq \|\boldsymbol{z}_i - C_{\mathcal{D}_i}\|$ for all $i \in \mathcal{I}$ then $\mathbb{J}(\boldsymbol{z}', \mathfrak{D}) \leq \mathbb{J}(\boldsymbol{z}, \mathfrak{D}(\boldsymbol{z}))$. Additionally, from Lemma 1 and 12 we know that $\mathbb{J}(\boldsymbol{z}, \mathfrak{R}(\boldsymbol{z})) \leq \mathbb{J}(\boldsymbol{z}, \mathfrak{D})$, and $\mathbb{J}(\mathbb{T}(\boldsymbol{z}, \boldsymbol{w}), \mathfrak{R}(\boldsymbol{z})) \leq \mathbb{J}(\boldsymbol{z}, \mathfrak{R}(\boldsymbol{z}))$.

(iii) Follows from the fact that the Lloyd map meets the specifications in Lemma 12.

■

Algorithm 3 Lloyd Algorithm [15]

Initialization: $\mathcal{U}$, ϱ, $\boldsymbol{z}^0 \in \Gamma$
Iteration: At time k, agent i performs:
Compute $\mathcal{R}_i^k$ using Algorithm 1
$\boldsymbol{z}_i^{k+1} = (\int_{\mathcal{R}_i^k} \varrho(\boldsymbol{q}) d\boldsymbol{q})^{-1} \int_{\mathcal{R}_i^k} \boldsymbol{q} \varrho(\boldsymbol{q}) d\boldsymbol{q}$

Chapter 5

Experimental Results

5.1 Experimental Setup

In this section we give an overview of the laboratory setup used to collect the experimental data [32], which is depicted in Figure 5.1. A fleet of BitCraze Crazyflies 2.0 [46] employing the local control policies and FES controllers specified in Chapters 3 and 4 is used to validate the theoretical findings of [1] for *formation control* and *optimal coverage*. Each Crazyflie is unambiguously identified by a unique triplet of passive markers mounted on its chassis, which allows accurate detection by an external motion capture system. Five motion capture cameras provide accurate measurements [47] of the position coordinates $\boldsymbol{p}_i^{\mathcal{O}}$ and the attitude of the body fixed frame $\boldsymbol{\psi}_i$ for each Crazyflie $i \in \mathcal{I}$ by running a specialized software on a dedicated machine, and delimit the admissible flying region. Measurement data is made available at 200Hz to off-board laptops, each one assigned to only one quadcopter, that compute the control of the FES controller $\boldsymbol{u}_i$ and the local controller $\boldsymbol{v}_i$ according to the scheme in Section 2.2. In particular, position $\boldsymbol{p}^{\mathcal{O}}$ and attitude $\boldsymbol{\psi}$ measurements include all the quadcopters that populate the flying region. Computation of $\boldsymbol{u}_i$ is carried on in an online, distributed fashion, i.e., including only measurements of the neighbouring Crazyflies in the solution algorithm. Finally, the vector $\boldsymbol{v}_i$ is sent to the *inner* controller where it is converted to rotation velocities of the propellers.

5.2 Performance Measures

Let $\overline{\boldsymbol{u}}_i^k$ denote the optimal placement of agent i and $h(\overline{\boldsymbol{u}}_i^k, \boldsymbol{w}^k)$ the corresponding steady-state output vector at time step $k \in \mathbb{Z}_+$. As a performance measure we consider the three-dimensional Euclidean distance $\|\boldsymbol{y}_i^k - h(\overline{\boldsymbol{u}}_i^k, \boldsymbol{w}^k)\|$ between the current position $\boldsymbol{y}_i^k$ of the i-th Crazyflie and $h(\overline{\boldsymbol{u}}_i^k, \boldsymbol{w}^k)$. In particular, we expect each Crazyflie to track its solution trajectory and eventually converge with zero error to the desired steady-state.
Experimental results are discussed in the next sections with illustrations obtained from real data.

5.3 Formation Control

We consider a swarm of five Crazyflies such that $M = 2$, $N = 3$. Recalling the problem formulation in Section 4.1, this means that the network includes two *masters* and three *followers*, forming an LTI interaction graph. In our experiments, $\mathcal{G}$ consists of a next-hop structure connecting the two elements of $\mathcal{M}$ using one and only one path that includes all $i \in \mathcal{I}$. Formally, this is the Hamiltonian path connecting the *master* Crazyflies and it meets all the specifications made by Assumption 6. The running experiment is the following: all *masters* move freely in

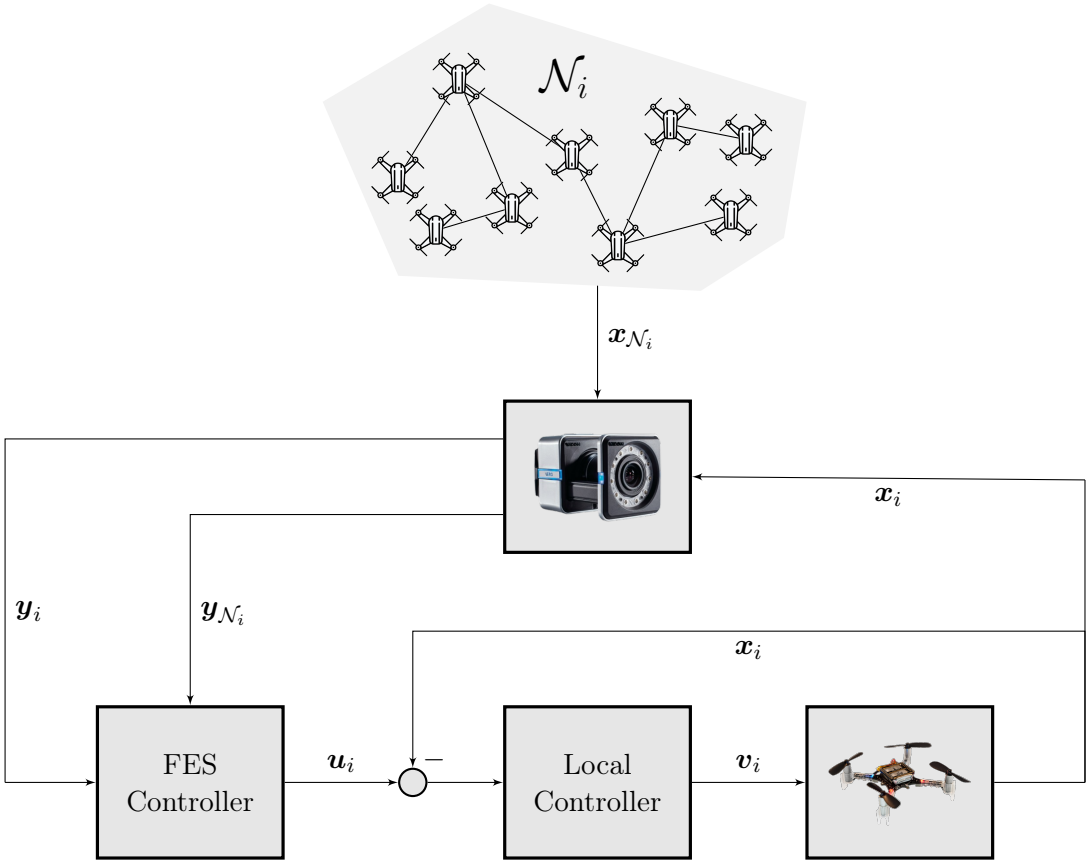


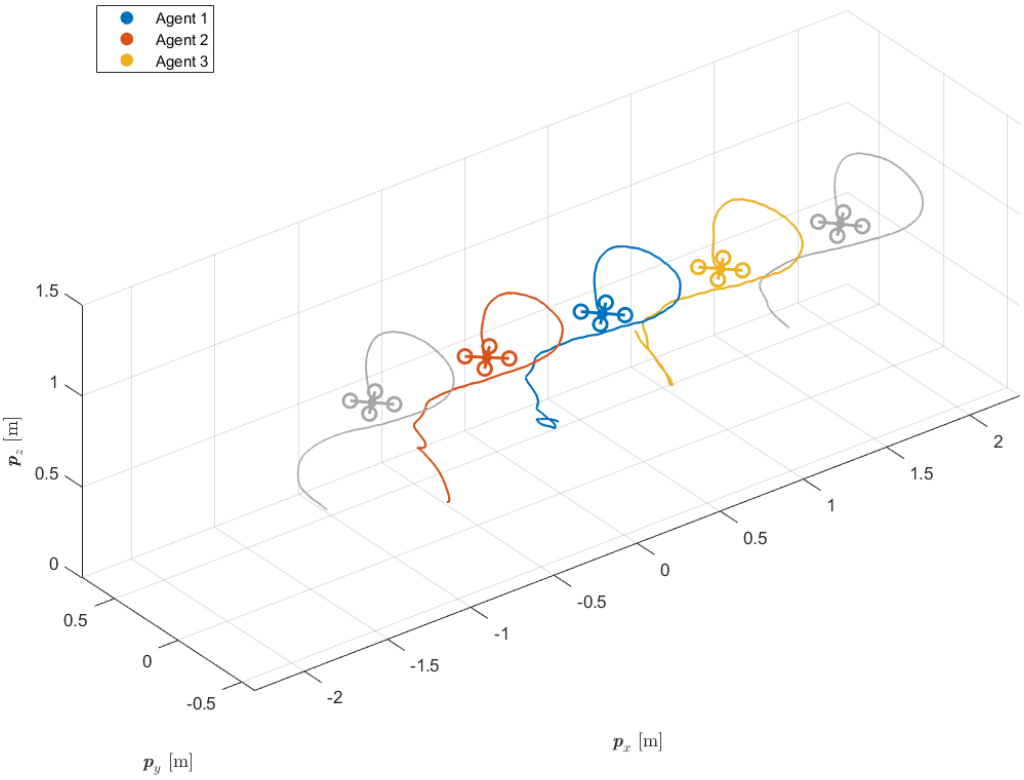
Figure 5.1: Block diagram showing the experimental setup for the i-th Crazyflie. Output measurements $\boldsymbol{y}_i$ and $\boldsymbol{y}_{-i}$ are obtained for i and the other agents $-i \in \mathcal{I} \setminus i$ in the flying space. In a distributed fashion, only a subset of the latter is fed to the ES algorithm in the FES controller, namely $\boldsymbol{y}_i$ and $\boldsymbol{y}_{\mathcal{N}_i}$.

the closed parallelepiped $\mathcal{FR}$ and entrust the three *followers* with establishing a reliable communication channel at all times. In particular, each element $j \in \mathcal{M}$ moves for a certain time interval $t \in [0, t_{\max}]$ after the start of experiment until it reaches a desired position $\overline{\boldsymbol{w}}_j \in \mathcal{W}_j$ at $t_{\max} = 12.5$. If $i \in \mathcal{I}$ is given measurements of $\boldsymbol{y}_i^k$, $\boldsymbol{y}_{\mathcal{N}_i}^k$ and $\boldsymbol{w}_j^k$ for all $j \in \mathcal{M} \cap \mathcal{N}_i$, then $\overline{\boldsymbol{u}}_i^{k+1}$ can be computed in closed form or after a sufficient number of partial gradient steps starting from $\boldsymbol{y}_i^k$. Due to the structure of its local utility $\mathbb{J}_i$, each *follower* attempts to minimize the Euclidean distance from its two neighbours, which has a unique minimum $h(\overline{\boldsymbol{u}}_i^k, \boldsymbol{w}^k)$ in the mid point of the segment joining the neighbours in $\mathcal{N}_i$. Globally, the optimal deployment is obtained when all *followers* position themselves on the straight line joining the *masters* at equal distances from each other. This behaviour is illustrated in Figure 5.2 and 5.3 for a fleet of LQR- and MPC-controlled Crazyflies that update their position setpoint $\boldsymbol{u}_i^k$ by taking one PPGD update at each time step. In our experiments, this is done at 200Hz. Precise convergence of $\|\boldsymbol{y}_i^k - h(\overline{\boldsymbol{u}}_i^k, \boldsymbol{w}^k)\|$ to a small neighbourhood of zero is obtained after the *masters* stop at $\overline{\boldsymbol{w}} = [\overline{\boldsymbol{w}}_j]_{j \in \mathcal{M}}$. The tracking performance is shown in Figure 5.4.

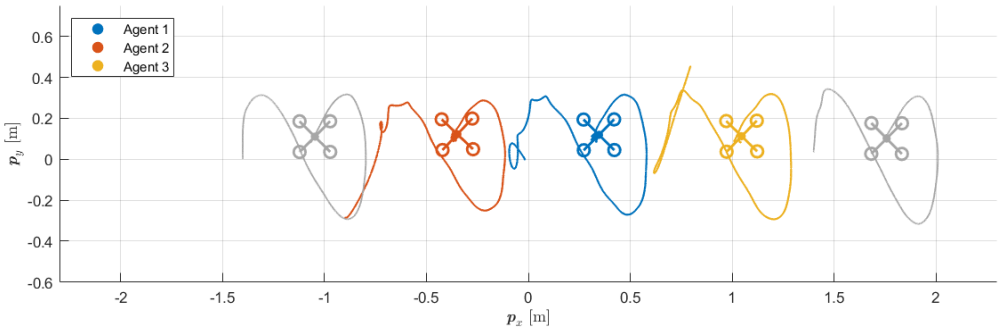
5.4 Optimal Coverage Problem

Let now $\mathcal{FR}$ be a compact parallelepiped, $\mathcal{I}$ a formation of $N = 4$ Crazyflies, and $\boldsymbol{p}^0$ their vector of stacked initial positions in inertial frame coordinates, where we omit the superscript $\mathcal{O}$ for convenience. As a running experiment for *optimal coverage* using FES we consider a

time-varying density function $\varrho(\boldsymbol{q}, t)$ spread onto $\mathcal{FR}$ to denote the Internet demand at location $\boldsymbol{q} \in \mathcal{FR}$ and time $t \in \mathbb{R}_+$. We allow ϱ to change in the time interval $[0, t_{\max}]$, where $t_{\max} = 25$. The Crazyflies are expected to track the moving distribution and converge to a centroidal Voronoi tessellation once the distribution stops moving. In our experiment, we considered a Gaussian density $\varrho(\boldsymbol{q}, t) \sim \mathcal{N}(\mu(t), \Sigma(t))$, where $\mu(t)$, $\Sigma(t)$ denote the time-varying mean vector and covariance matrix. At each time step, of duration $\tau = 5$, the control action $\boldsymbol{u}_i^k \in \mathcal{FR}$ is computed after one iteration of (4.14). Instead, $\overline{\boldsymbol{u}}_i^k$ and $h(\overline{\boldsymbol{u}}_i^k)$ are obtained after ten Lloyd updates of the current output vector $\boldsymbol{y}^k$, used as an initial guess. Requiring evaluation of many multi-dimensional integrals and of the Voronoi regions $\mathcal{V}_i^k$, the Lloyd iteration is very demanding in terms of computational resources, hence justifying the use of large time steps. Figure 5.5 shows the tracking performance when nominal MPC is considered as the local controller.
To conclude, we refer the interested reader to [48] and [49] for a thorough documentation on the specific libraries that were adopted in order to distributedly compute Voronoi diagrams and perform Montecarlo integration.

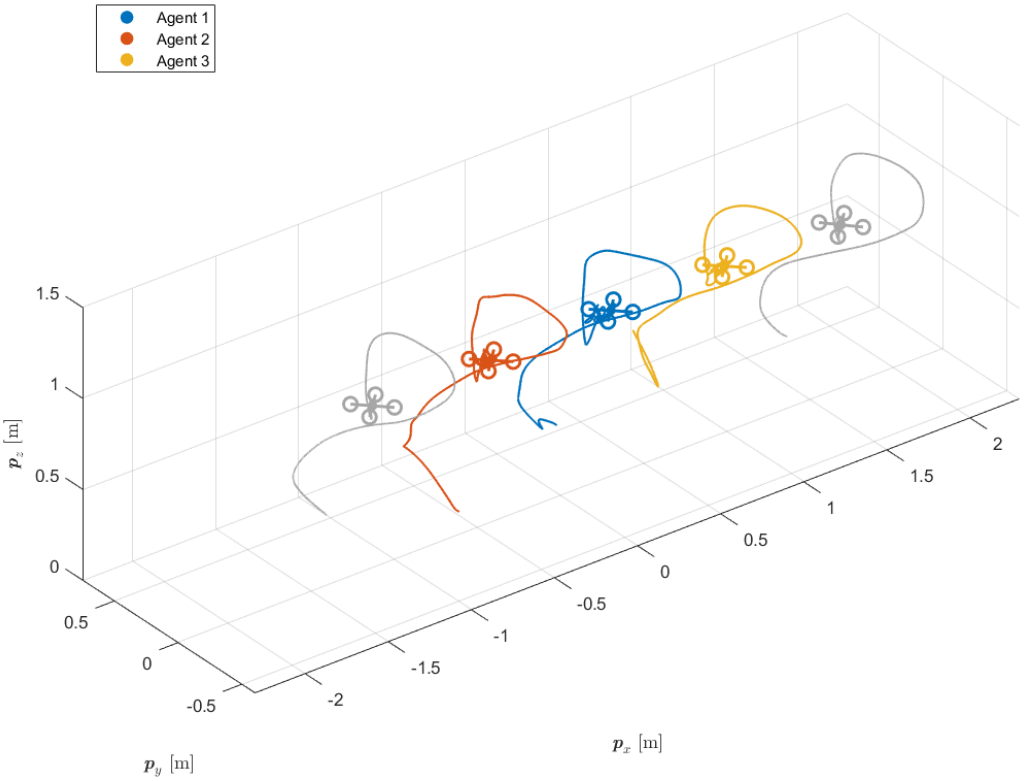


(a)

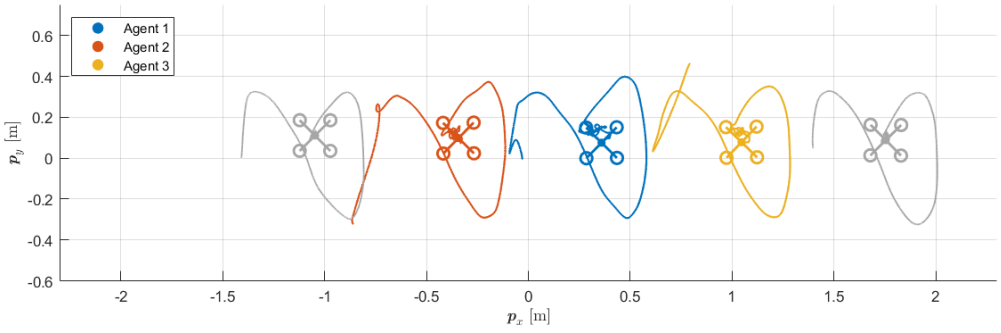


(b)

Figure 5.2: State trajectory of three *followers* implementing LQR as a local controller and projected pseudo-gradient descent as the ES algorithm. Convergence to the expected straight line configuration joining the *masters*, which are depicted in gray, is attained by the proposed controller.

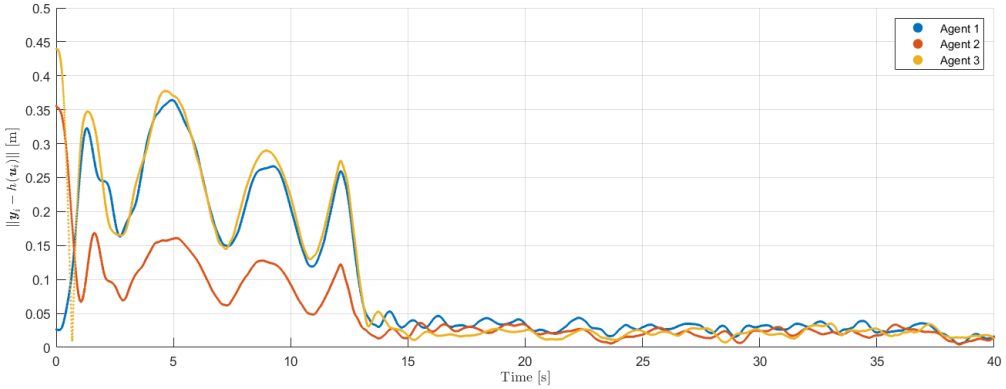


(a)

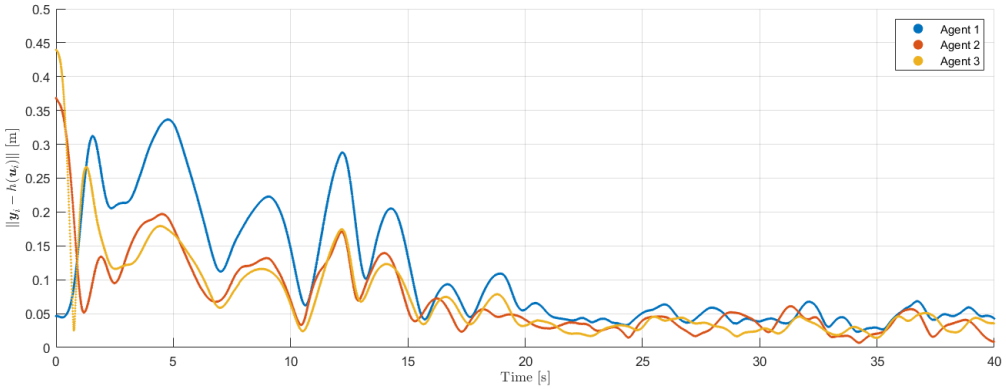


(b)

Figure 5.3: State trajectory resulting from the interconnection of the MPC-controlled Crazyflies with projected pseudo-gradient descent. The *master* nodes follow the same state trajectories as in Fig 5.2. Again, Fig. 5.3a and 5.3b illustrate convergence to the optimal configuration, which is unknown when the experiment starts.



(a)



(b)

Figure 5.4: Convergence to the optimal deployment for the LQR- and MPC-controlled physical system is shown in Figure 5.4a and 5.4b, respectively. Larger oscillations in $\|\boldsymbol{y}_i^k - h(\overline{\boldsymbol{u}}_i^k, \boldsymbol{w}^k)\|$ are observed for $t \leq t_{\max}$, i.e., during the time interval in which the *masters* are allowed to move. We note that LQR is more reactive than nominal MPC due to the lack of constraints in the state and control input variables.

Figure 5.5: Optimal coverage control: tracking performance.

Chapter 6

Conclusions

This work constitutes the first successful validation of Feedback Equilibrium Seeking for three-dimensional coordination of UAV-aided networks. Two discrete-time iterative controllers were proposed that attain the optimal deployment of a fleet of Bitcraze Crazyflies in dynamic environments. *Formation* and *coverage control* have been analyzed under the FES lens and their effectiveness when modelling cooperation among multiple quadcopters assessed for popular applications such as mobile relay networks. A robust stability analysis of the closed-loop between the algorithm and the pre-stabilized model of the quadcopters was given. Local input-to-state stability of the sampled-data system with respect to fluctuations in the position of the *masters* Crazyflies or in the density function was proved through verification of the specifications in [1]. The theoretical findings were then validated on a fleet or real-world quadcopters.
Future research would focus on deriving sufficient conditions on the density functions to ensure existence of finitely many optimal network configurations for *coverage* control and comparing performance to other types of controllers, e.g., distributed sub-optimal MPC [35].

Bibliography

[1] Giuseppe Belgioioso et al. "Online Feedback Equilibrium Seeking". In: *arXiv preprint arXiv:2210.12088* (2022).

[2] Chris A Wargo et al. "Unmanned Aircraft Systems (UAS) research and future analysis". In: *2014 IEEE Aerospace conference*. IEEE. 2014, pp. 1–16.

[3] Mbazingwa Elirehema Mkiramweni et al. "A Survey of Game Theory in Unmanned Aerial Vehicles Communications". In: *IEEE Communications Surveys & Tutorials* 21.4 (2019), pp. 3386–3416.

[4] Ramesh Chembil Palat, A Annamalau, and JR Reed. "Cooperative relaying for ad-hoc ground networks using swarm UAVs". In: *MILCOM 2005-2005 IEEE Military Communications Conference*. IEEE. 2005, pp. 1588–1594.

[5] Pengcheng Zhan, Kai Yu, and A. Lee Swindlehurst. "Wireless Relay Communications with Unmanned Aerial Vehicles: Performance and Optimization". In: *IEEE Transactions on Aerospace and Electronic Systems* 47.3 (2011), pp. 2068–2085.

[6] Jorge Cortes et al. "Coverage control for mobile sensing networks". In: *IEEE Transactions on robotics and Automation* 20.2 (2004), pp. 243–255.

[7] Imad Jawhar, Nader Mohamed, and Jameela Al-Jaroodi. "UAV-based data communication in wireless sensor networks: Models and strategies". In: *2015 International conference on unmanned aircraft systems (ICUAS)*. IEEE. 2015, pp. 687–694.

[8] Wenhao Luo and Katia Sycara. "Voronoi-based coverage control with connectivity maintenance for robotic sensor networks". In: *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE. 2019, pp. 148–154.

[9] So-Yeon Park et al. "DroneNetX: Network Reconstruction Through Connectivity Probing and Relay Deployment by Multiple UAVs in Ad Hoc Networks". In: *IEEE Transactions on Vehicular Technology* 67.11 (2018), pp. 11192–11207.

[10] Hamid Menouar et al. "UAV-enabled intelligent transportation systems for the smart city: Applications and challenges". In: *IEEE Communications Magazine* 55.3 (2017), pp. 22–28.

[11] Hassan Jaleel and Jeff S Shamma. "Distributed optimization for robot networks: From real-time convex optimization to game-theoretic self-organization". In: *Proceedings of the IEEE* 108.11 (2020), pp. 1953–1967.

[12] Yuan Yan. *Co-Optimization of Communication, Motion and Sensing in Mobile Robotic Operations*. University of California, Santa Barbara, 2016.

[13] Yasamin Mostofi, Mehrzad Malmirchegini, and Alireza Ghaffarkhah. "Estimation of communication signal strength in robotic networks". In: *2010 IEEE International Conference on Robotics and Automation*. 2010, pp. 1946–1951.

[14] Jorge Cortés et al. "Coverage control for mobile sensing networks: Variations on a theme". In: *Mediterranean Conference on Control and Automation*. Lisbon, Portugal Lisbon, Portugal. 2002, pp. 9–13.

[15] Qiang Du, Maria Emelianenko, and Lili Ju. "Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations". In: *SIAM journal on numerical analysis* 44.1 (2006), pp. 102–119.

[16] Yang Liu et al. "On centroidal Voronoi tessellation—energy smoothness and fast computation". In: *ACM Transactions on Graphics (ToG)* 28.4 (2009), pp. 1–17.

[17] Adrian Hauswirth et al. "Optimization algorithms as robust feedback controllers". In: *arXiv preprint arXiv:2103.11329* (2021).

[18] Antonio Terpin et al. "Distributed feedback optimisation for robotic coordination". In: *2022 American Control Conference (ACC)*. IEEE. 2022, pp. 3710–3715.

[19] Philip B Charlesworth. "A non-cooperative game to coordinate the coverage of two communications UAVs". In: *MILCOM 2013-2013 IEEE Military Communications Conference*. IEEE. 2013, pp. 668–673.

[20] Philip B Charlesworth and Stuart M Allen. "Use of dynamic flight paths to enhance support to priority subscribers on a communications UAV". In: *MILCOM 2012-2012 IEEE Military Communications Conference*. IEEE. 2012, pp. 1–6.

[21] Julian Barreiro-Gomez et al. "Distributed formation control of multiple unmanned aerial vehicles over time-varying graphs using population games". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 5245–5250.

[22] Shuhang Zhang et al. "Joint trajectory and power optimization for UAV relay networks". In: *IEEE Communications Letters* 22.1 (2017), pp. 161–164.

[23] Xijian Zhong et al. "Joint optimization of relay deployment, channel allocation, and relay assignment for UAVs-aided D2D networks". In: *IEEE/ACM Transactions on Networking* 28.2 (2020), pp. 804–817.

[24] Giuseppe Belgioioso et al. "Sampled-data online feedback equilibrium seeking: Stability and tracking". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 2702–2708.

[25] Qiang Du, Vance Faber, and Max Gunzburger. "Centroidal Voronoi tessellations: Applications and algorithms". In: *SIAM review* 41.4 (1999), pp. 637–676.

[26] Steven Fortune. "Voronoi diagrams and Delaunay triangulations". In: *Computing in Euclidean geometry* (1995), pp. 225–265.

[27] Rolf Klein. *Concrete and abstract Voronoi diagrams*. Vol. 400. Springer Science & Business Media, 1989.

[28] Paul N Beuchat. "N-rotor vehicles: modelling, control, and estimation". In: (2019).

[29] Sergei Lupashin et al. "A platform for aerial robotics research and demonstration: The flying machine arena". In: *Mechatronics* 24.1 (2014), pp. 41–54.

[30] Robert Mahony, Vijay Kumar, and Peter Corke. "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor". In: *IEEE robotics & automation magazine* 19.3 (2012), pp. 20–32.

[31] Kyaw Myat Thu and AI Gavrilov. "Designing and modeling of quadcopter control system using L1 adaptive control". In: *Procedia Computer Science* 103 (2017), pp. 528–535.

[32] Ezzat Elokda et al. "Data-enabled predictive control for quadcopters". In: *International Journal of Robust and Nonlinear Control* 31.18 (2021), pp. 8916–8936.

[33] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*. john Wiley & sons, 2005.

[34] Francesco Bullo. *Lectures on network systems*. Vol. 1. Kindle Direct Publishing, 2020.

[35] Giuseppe Belgioioso et al. "Stability and Robustness of Distributed Suboptimal Model Predictive Control". In: *arXiv preprint arXiv:2211.07341* (2022).

[36] Graham Goodwin, María M Seron, and José A De Doná. *Constrained control and estimation: an optimisation approach*. Springer Science & Business Media, 2006.

[37] Alberto Bemporad et al. "The explicit linear quadratic regulator for constrained systems". In: *Automatica* 38.1 (2002), pp. 3–20.

[38] Panagiotis Patrinos and Haralambos Sarimveis. "Convex parametric piecewise quadratic optimization: Theory and algorithms". In: *Automatica* 47.8 (2011), pp. 1770–1777.

[39] Michael Sabin and Robert Gray. "Global convergence and empirical consistency of the generalized Lloyd algorithm". In: *IEEE Transactions on information theory* 32.2 (1986), pp. 148–155.

[40] Giuseppe Belgioioso et al. "Distributed generalized Nash equilibrium seeking: An operator-theoretic perspective". In: *IEEE Control Systems Magazine* 42.4 (2022), pp. 87–102.

[41] Giuseppe Belgioioso and Sergio Grammatico. "Projected-gradient algorithms for generalized equilibrium seeking in aggregative games arepreconditioned forward-backward methods". In: *2018 European Control Conference (ECC)*. IEEE. 2018, pp. 2188–2193.

[42] Giuseppe Belgioioso and Sergio Grammatico. "Semi-Decentralized Generalized Nash Equilibrium Seeking in Monotone Aggregative Games". In: *IEEE Transactions on Automatic Control* 68.1 (2023), pp. 140–155.

[43] Heinz H Bauschke, Patrick L Combettes, et al. *Convex analysis and monotone operator theory in Hilbert spaces*. Vol. 408. Springer, 2011.

[44] Zvi Drezner. *Facility location: a survey of applications and methods*. Springer, 1995.

[45] Walter Rudin et al. *Principles of mathematical analysis*. Vol. 3. McGraw-Hill, 1976.

[46] AB Bitcraze. *Multi-ranger deck*. 2019.

[47] Pierre Merriaux et al. "A study of vicon system positioning performance". In: *Sensors* 17.7 (2017), p. 1591.

[48] Chris Rycroft. *Voro++: A three-dimensional Voronoi cell library in C++*. Tech. rep. Lawrence Berkeley National Lab. (LBNL), Berkeley, CA (United States), 2009.

[49] Mark Galassi et al. *GNU scientific library*. Network Theory Limited Godalming, 2002.

[50] Masao Iri, Kazuo Murota, and Takao Ohya. "A fast Voronoi-diagram algorithm with applications to geographical optimization problems". In: *System Modelling and Optimization: Proceedings of the 11th IFIP Conference Copenhagen*. Springer. 1984, pp. 273–288.

Appendix A

System Modelling

A.1 Linearized Equations of Motion

In Section 2.1 we acknowledged that the equilibrium inputs are identical for each position and yaw angle, hence making hovering the only relevant operating condition for a quadcopter. Here we derive the linearized EoMs for (2.1), (2.2) about the equilibrium $\left(\overline{\tilde{\boldsymbol{x}}},\ \overline{\tilde{\boldsymbol{v}}}\right)$. To simplify notation we will not specify the reference frames in which the quantities are expressed.

Let $\tilde{\boldsymbol{v}}' = (mg\ 0\ 0\ 0)^\top$ be a control input applied to the non-linear EoMs. In the absence of external disturbances, total thrust opposing gravity and zero net torque about the body-frame define the unique control input allowing hovering at a fixed position. Thus $\overline{\tilde{\boldsymbol{v}}} = \tilde{\boldsymbol{v}}'$. Let again $\boldsymbol{f}_i$ be the thrust force acting at the position $\boldsymbol{p}_i$ of propeller i: $\overline{\tilde{\boldsymbol{v}}}$ is attained by all solutions to the linear system of equations

$$\overline{\tilde{\boldsymbol{v}}} = \begin{pmatrix} mg \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 1 \\ p_{1,y} & \cdots & p_{4,y} \\ -p_{1,x} & \cdots & -p_{4,x} \\ c & \cdots & c \end{pmatrix} \begin{pmatrix} f_{1,z} \\ f_{2,z} \\ f_{3,z} \\ f_{4,z} \end{pmatrix}. \tag{A.1}$$

Under $\overline{\tilde{\boldsymbol{v}}}$, the quadcopter will fly in a fixed position with an arbitrary yaw orientation. Therefore $\overline{\tilde{\boldsymbol{x}}} = \left(\overline{\boldsymbol{p}}\ \ \boldsymbol{0}_{1\times 5}\ \ \overline{\alpha}\ \ \boldsymbol{0}_{1\times 3}\right)^\top$ is an equilibrium point for the non-linear system for all $\overline{\boldsymbol{p}} \in \mathbb{R}^3$, $\alpha \in \mathbb{R}$. The linearized system writes as

$$\begin{pmatrix} \delta\dot{\boldsymbol{p}} \\ \delta\ddot{\boldsymbol{p}} \\ \delta\dot{\psi} \\ \delta\dot{\boldsymbol{\omega}} \end{pmatrix} = \left.\begin{pmatrix} \frac{\partial}{\partial \boldsymbol{p}}\dot{\boldsymbol{p}} & \frac{\partial}{\partial \dot{\boldsymbol{p}}}\dot{\boldsymbol{p}} & \frac{\partial}{\partial \psi}\dot{\boldsymbol{p}} & \frac{\partial}{\partial \boldsymbol{\omega}}\dot{\boldsymbol{p}} \\ \frac{\partial}{\partial \boldsymbol{p}}\ddot{\boldsymbol{p}} & \frac{\partial}{\partial \dot{\boldsymbol{p}}}\ddot{\boldsymbol{p}} & \frac{\partial}{\partial \psi}\ddot{\boldsymbol{p}} & \frac{\partial}{\partial \boldsymbol{\omega}}\ddot{\boldsymbol{p}} \\ \frac{\partial}{\partial \boldsymbol{p}}\dot{\psi} & \frac{\partial}{\partial \dot{\boldsymbol{p}}}\dot{\psi} & \frac{\partial}{\partial \psi}\dot{\psi} & \frac{\partial}{\partial \boldsymbol{\omega}}\dot{\psi} \\ \frac{\partial}{\partial \boldsymbol{p}}\dot{\boldsymbol{\omega}} & \frac{\partial}{\partial \dot{\boldsymbol{p}}}\dot{\boldsymbol{\omega}} & \frac{\partial}{\partial \psi}\dot{\boldsymbol{\omega}} & \frac{\partial}{\partial \boldsymbol{\omega}}\dot{\boldsymbol{\omega}} \end{pmatrix}\right|_{\substack{\tilde{\boldsymbol{x}} = \overline{\tilde{\boldsymbol{x}}} \\ \tilde{\boldsymbol{v}} = \overline{\tilde{\boldsymbol{v}}}}} \begin{pmatrix} \delta\boldsymbol{p} \\ \delta\dot{\boldsymbol{p}} \\ \delta\psi \\ \delta\boldsymbol{\omega} \end{pmatrix} + \left.\begin{pmatrix} \frac{\partial}{\partial f_z}\dot{\boldsymbol{p}} & \frac{\partial}{\partial \boldsymbol{\tau}}\dot{\boldsymbol{p}} \\ \frac{\partial}{\partial f_z}\ddot{\boldsymbol{p}} & \frac{\partial}{\partial \boldsymbol{\tau}}\ddot{\boldsymbol{p}} \\ \frac{\partial}{\partial f_z}\dot{\psi} & \frac{\partial}{\partial \boldsymbol{\tau}}\dot{\psi} \\ \frac{\partial}{\partial f_z}\dot{\boldsymbol{\omega}} & \frac{\partial}{\partial \boldsymbol{\tau}}\dot{\boldsymbol{\omega}} \end{pmatrix}\right|_{\substack{\tilde{\boldsymbol{x}} = \overline{\tilde{\boldsymbol{x}}} \\ \tilde{\boldsymbol{v}} = \overline{\tilde{\boldsymbol{v}}}}} \begin{pmatrix} \delta f_z \\ \delta\boldsymbol{\tau} \end{pmatrix},$$

or in short

$$\delta\dot{\tilde{\boldsymbol{x}}} = \tilde{A}\delta\tilde{\boldsymbol{x}} + \tilde{B}\delta\tilde{\boldsymbol{v}}, \tag{A.2}$$

where $\delta(\cdot)$ indicates the deviation of $(\cdot)$ from $\left(\overline{\tilde{\boldsymbol{x}}},\ \overline{\tilde{\boldsymbol{v}}}\right)$. Note that, if a controller is designed based on the linearized model, the control input that will drive the system to the desired equilibrium is $\tilde{\boldsymbol{v}} = \overline{\tilde{\boldsymbol{v}}} + \delta\tilde{\boldsymbol{v}}$ and that the decouplings in the dynamics cause many blocks of the $\tilde{A}$ and $\tilde{B}$ to vanish, regardless of our choice of $\overline{\tilde{\boldsymbol{x}}}$.

A.2 Robust Stability of a Quadcopter

Proof of Lemma 5.

(i) Let $\boldsymbol{x}', \boldsymbol{x}'' \in \mathbb{R}^{n_x}$ and $\boldsymbol{u} \in \mathbb{R}^{n_u}$. Since $f\colon (\mathbb{R}^{n_x} \times \mathbb{R}^{n_u}, \mathbb{R}, \|\cdot\|) \to (\mathbb{R}^{n_x}, \mathbb{R}, \|\cdot\|)$ is a linear function between finite dimensional normed spaces, it is also continuous. Global Lipschitz continuity of f with respect to $\boldsymbol{x}$ follows from the properties of induces norms of continuous functions:

$$\begin{aligned}\|f(\boldsymbol{x}'', p(\boldsymbol{u})) - f(\boldsymbol{x}', p(\boldsymbol{u}))\| &= \|(A - BK)(\boldsymbol{x}'' - p(\boldsymbol{u})) - (A - BK)(\boldsymbol{x}' - p(\boldsymbol{u}))\| \\ &\leq \|A - BK\| \|\boldsymbol{x}'' - \boldsymbol{x}'\|.\end{aligned}$$

The same procedure applies to show that f is globally Lipschitz in $\boldsymbol{u}$.

(ii) Let again $\boldsymbol{x}', \boldsymbol{x}'' \in \mathbb{R}^{n_x}$ and $\boldsymbol{u} \in \mathbb{R}^{n_u}$. Global Lipschitz continuity in $\boldsymbol{x}$ follows trivially from continuity of the linear map g

$$\|g(\boldsymbol{x}'') - g(\boldsymbol{x}')\| \leq \|C\| \, \|\boldsymbol{x}'' - \boldsymbol{x}'\|.$$

(iii) $\mathcal{U} = \mathcal{F}\mathcal{R}$ is a compact polytope by design.

(iv) The following lemma gives conditions under which the closed-loop system with the state-feedback (3.3) is globally asymptotically stable. This is not surprising as stability for linear systems is always global.

Lemma 13 (Lyapunov Function for LQR)**.** *If $\big(A, B\big)$ is stabilizable and $\big(Q^{1/2}, A\big)$ is detectable, then the optimal value function $V(\boldsymbol{x}, \boldsymbol{u}) = \|\boldsymbol{x} - C\boldsymbol{u}\|_P^2$ is a Lyapunov function for (3.2a), P is the unique positive definite solution of (3.4) and K is the feedback policy that minimizes the value function.*

All conditions are met, thus (3.2) is globally asymptotically stable and local exponential stability of f follows from Lemma 4. We can write the following inequalities for V

$$\underbrace{\lambda_{\min}(P)}_{\alpha_1} \|\boldsymbol{x} - p(\boldsymbol{u})\|^2 \leq \|\boldsymbol{x} - p(\boldsymbol{u})\|_P^2 \leq \underbrace{\lambda_{\max}(P)}_{\alpha_2} \|\boldsymbol{x} - p(\boldsymbol{u})\|^2$$

and for $\dot{V}$

$$\begin{aligned}\dot{V}(\boldsymbol{x}(t), \boldsymbol{u}) &= \|\boldsymbol{x}(t) - p(\boldsymbol{u})\|^2_{(A-BK)^\top P + P(A-BK)} \\ &= -\|\boldsymbol{x}(t) - p(\boldsymbol{u})\|^2_{Q + PBR^{-1}B^\top P} \\ &\leq -\underbrace{\frac{\lambda_{\min}(S)}{\lambda_{\max}(P)}}_{\alpha_3} \|\boldsymbol{x}(t) - p(\boldsymbol{u})\|_P^2\end{aligned}$$

where $S = Q + PBR^{-1}B^\top P \in \mathbb{S}^{n_x}_{++}$ form the controllability and observability assumptions.

Stability of the non-linear EoMs is ensured by Assumption 4 and Lemma 4. ■

Proof of Lemma 7.

(i) Let $\boldsymbol{x}', \boldsymbol{x}'' \in \mathcal{X}_0$, $\boldsymbol{u} \in \mathcal{U}$. Global Lipschitz continuity of (3.8a) in $\boldsymbol{x}$ follows from the fact that it is piecewise affine

$$\|f(\boldsymbol{x}'', \boldsymbol{u}) - f(\boldsymbol{x}', \boldsymbol{u})\| \leq \left(\max_{\mathcal{CR}_j \in \mathcal{X}_0} \|\hat{A} + \hat{B}K_j\| \right) \|\boldsymbol{x}'' - \boldsymbol{x}'\|.$$

(ii) Global Lipschitz continuity follows from linearity of (3.8b)

$$\|g(\boldsymbol{x}'') - g(\boldsymbol{x}')\| \leq \|\hat{C}\| \|\boldsymbol{x}'' - \boldsymbol{x}'\|.$$

(iii) $\mathcal{U}$ is compact and convex by design.

(iv) Exponential stability of the MPC-controlled plant is proved in [36, §5.6.3]. Let $\delta\mathfrak{X}^*(\delta\boldsymbol{x})$, $\delta\mathfrak{V}^*(\delta\boldsymbol{x})$ be the optimal state sequence and control input sequence expressed as a function of the initial conditions: the following is a valid Lyapunov function

$$V(\delta\boldsymbol{x}) = \|\delta\boldsymbol{x}_J^*\|_P^2 + \sum_{j=0}^{J-1} \left(\|\delta\boldsymbol{x}_j^*\|_Q^2 + \|\delta\boldsymbol{v}_j^*\|_R^2 \right).$$

For simplicity, in the rest of the proof we use a slightly different notation than in [1]. Since V is a sum of positive definite forms the following inequality holds

$$V(\delta\boldsymbol{x}) \geq \|\delta\boldsymbol{x}\|_Q^2 \geq \underbrace{\lambda_{\min}(Q)}_{\alpha_1} \|\delta\boldsymbol{x}\|^2.$$

Take now $\delta\mathfrak{V}$ to be the feasible sequence of all zero inputs. Note that this choice of $\delta\mathfrak{V}$ is sub-optimal and possibly leading to an unfeasible state evolution $\delta\mathfrak{X}$. It holds that

$$V(\delta\boldsymbol{x}) \leq \|\hat{A}^{J-1}\delta\boldsymbol{x}\|_P^2 + \sum_{j=0}^{J-2} \|\hat{A}^j\delta\boldsymbol{x}\|_Q^2 \leq \underbrace{\lambda_{\max}\left((\hat{A}^{J-1})^\top P \hat{A}^{J-1} + \sum_{j=0}^{J-2} (A^j)^\top Q A^j \right)}_{\alpha_2} \|\delta\boldsymbol{x}\|^2.$$

The following is an upper bound on the energy decrease condition:

$$V(\delta\boldsymbol{x}_{k+1}) - V(\delta\boldsymbol{x}_k) \leq -\|\delta\boldsymbol{x}_k\|_Q^2 - \|\delta\boldsymbol{v}^*\|_R^2 \leq -\|\delta\boldsymbol{x}_k\|_Q^2 \leq -\underbrace{\lambda_{\min}(Q)}_{\alpha_3} \|\delta\boldsymbol{x}_k\|^2.$$

■

Appendix B

The Optimal Coverage Problem

B.1 Computation of Gradients

In contrast to [6, 25, 50] this section provides a self-contained derivation of (4.10). The notation used in Subsection 1.4.3 will be slightly modified according to that of Section 4.2, and should be restricted to the proof below to avoid confusion.

Proof.
Recall that $\boldsymbol{z} \in \mathbb{R}^{N_u}$ denotes the stacked positions of the agents, $\boldsymbol{e} \in \mathbb{R}^{Mn_u}$ the collection of vertices of the Voronoi diagram $\mathfrak{R}(\boldsymbol{z})$ induced by $\boldsymbol{z}$ on $\mathcal{FR}$. Using the chain rule in (4.11) we get

$$\nabla_{\boldsymbol{z}} \mathbb{J}(\boldsymbol{z}, \mathfrak{R}(\boldsymbol{z})) = \underbrace{\nabla_{\boldsymbol{z}} E(\boldsymbol{z})}_{\mathbf{a}} \underbrace{\nabla_{\boldsymbol{e}} \mathbb{J}(\boldsymbol{z}, \mathfrak{R}(\boldsymbol{e}))}_{\mathbf{b}} + \overbrace{\nabla_{\boldsymbol{z}} \mathbb{J}(\boldsymbol{z}, \mathfrak{R}(\boldsymbol{e}))|_{\boldsymbol{e} = E(\boldsymbol{z})}}^{\mathbf{c}}. \tag{B.1}$$

Lemma 14 (Reynolds Transport Theorem [25])**.** *Let $\Omega(\boldsymbol{x})$ be a smooth region in the variable $\boldsymbol{x}$, with a well defined boundary $\partial\Omega$. If $\tilde{f}(\boldsymbol{x}) = \int_{\Omega(\boldsymbol{x})} f(\boldsymbol{q}) d\boldsymbol{q}$, then it holds that*

$$\nabla_{\boldsymbol{x}} \tilde{f}(\boldsymbol{x}) = \int_{\partial\Omega(\boldsymbol{x})} f(\boldsymbol{q}) \nabla_{\boldsymbol{x}} \boldsymbol{q} \cdot \boldsymbol{n} d\boldsymbol{q},$$

with $\boldsymbol{n}$ the outward normal of $\partial\Omega$.

We tackle **b** first: from linearity of the gradient and Lemma 14 we obtain

$$\begin{aligned} \nabla_{\boldsymbol{e}} \mathbb{J}(\boldsymbol{z}, \mathfrak{R}(\boldsymbol{e})) &= \nabla_{\boldsymbol{e}} \left(\sum_{i \in \mathcal{I}} \int_{\mathcal{R}_i(\boldsymbol{e})} \|\boldsymbol{z}_i - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q} \right) \\ &= \sum_{i \in \mathcal{I}} \left(\nabla_{\boldsymbol{e}} \int_{\mathcal{R}_i(\boldsymbol{e})} \|\boldsymbol{z}_i - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q} \right) \\ &= \sum_{i \in \mathcal{I}} \left(\int_{\partial\mathcal{R}_i(\boldsymbol{e})} \Big(\|\boldsymbol{z}_i - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) \Big) \nabla_{\boldsymbol{e}} \boldsymbol{q}^\top \boldsymbol{n} d\boldsymbol{q} \right) \end{aligned} \tag{B.2}$$

where $\nabla_{\boldsymbol{e}} \boldsymbol{q}^\top \in \mathbb{R}^{n_u \times Mn_u}$ is a Jacobian, $\partial\mathcal{R}_i = \bigcup_{j \in \mathcal{N}_i} \Delta_i^j$ is the boundary of $\mathcal{R}_i$, where $\Delta_i^j \subset \mathcal{H}_i^j$ yields the equality in (1.9) for each $j \in \mathcal{N}_i$. We note that two neighbouring regions $\mathcal{R}_i$ and $\mathcal{R}_j$ have a common face Δ_i^j, but opposite normals $\boldsymbol{n}_i^j = -\boldsymbol{n}_j^i$. It is more convenient to write (B.2)

in terms of the Delaunay triangulation $\mathfrak{T}_{\boldsymbol{z}} = (\mathcal{I}, \mathcal{E}_{\mathfrak{T}})$.

$$\begin{aligned}
\nabla_{\boldsymbol{e}} \mathbb{J}(\boldsymbol{z}, \mathfrak{R}(\boldsymbol{e})) &= \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{N}_i} \int_{\Delta_i^j} (\|\boldsymbol{z}_i - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q})) \nabla_{\boldsymbol{e}} \boldsymbol{q}^\top \boldsymbol{n}_i^j d\boldsymbol{q} \\
&= \sum_{\{i,j\} \in \mathcal{E}_{\mathfrak{T}}} \left(\int_{\Delta_i^j} \Big(\|\boldsymbol{z}_i - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) \Big) \nabla_{\boldsymbol{e}} \boldsymbol{q}^\top \boldsymbol{n}_i^j d\boldsymbol{q} + \int_{\Delta_j^i} \Big(\|\boldsymbol{z}_j - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) \Big) \nabla_{\boldsymbol{e}} \boldsymbol{q}^\top \boldsymbol{n}_j^i d\boldsymbol{q} \right) \\
&= \sum_{\{i,j\} \in \mathcal{E}_{\mathfrak{T}}} \int_{\Delta_i^j} ((\|\boldsymbol{z}_i - \boldsymbol{q}\|^2 - \|\boldsymbol{z}_j - \boldsymbol{q}\|^2) \varrho(\boldsymbol{q})) \nabla_{\boldsymbol{e}} \boldsymbol{q}^\top \boldsymbol{n}_i^j d\boldsymbol{q} \\
&\overset{(1)}{=} \sum_{\{i,j\} \in \mathcal{E}_{\mathfrak{T}}} \int_{\Delta_i^j \in \partial \mathcal{F}\mathcal{R}} \big(\|\boldsymbol{z}_i - \boldsymbol{q}\|^2 - \|\boldsymbol{z}_j - \boldsymbol{q}\|^2 \big) \varrho(\boldsymbol{q}) \nabla_{\boldsymbol{e}} \boldsymbol{q}^\top \boldsymbol{n}_i^j d\boldsymbol{q} \overset{(2)}{=} 0
\end{aligned}$$

where (1) follows from $\|\boldsymbol{z}_i - \boldsymbol{q}\|^2 = \|\boldsymbol{z}_j - \boldsymbol{q}\|^2$, for all $\boldsymbol{q} \in \Delta_i^j$ such that Δ_i^j is not a subset of $\partial \mathcal{F}\mathcal{R}$, and (2) from $\partial \mathcal{F}\mathcal{R}$ being constant and hence independent of changes in $\boldsymbol{e}$. Substituting the previous equation inside (B.1) gives the expected result:

$$\begin{aligned}
\nabla_{\boldsymbol{z}} \mathbb{J}(\boldsymbol{z}, \mathfrak{R}(\boldsymbol{z})) &= \nabla_{\boldsymbol{z}} \mathbb{J}(\boldsymbol{z}, \mathfrak{R}(\boldsymbol{e}))\big|_{\boldsymbol{e} \,=\, E(\boldsymbol{z})} \\
&= \sum_{i \in \mathcal{I}} \left(\nabla_{\boldsymbol{z}} \int_{\mathcal{R}_i(\boldsymbol{e})} \|\boldsymbol{z}_i - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q} \right) \Bigg|_{\boldsymbol{e} \,=\, E(\boldsymbol{z})} \\
&= \left[2\boldsymbol{z}_i \int_{\mathcal{R}_i} \varrho(\boldsymbol{q}) d\boldsymbol{q} - 2 \int_{\mathcal{R}_i} \boldsymbol{q} \varrho(\boldsymbol{q}) d\boldsymbol{q} \right]_{i \in \mathcal{I}}.
\end{aligned}$$

which can be computed in a distributed way. ■

B.2 Coverage Control as a Potential Game

Let $\mathcal{I}$ be a player set, $\mathcal{U} = (\mathcal{U}_i)_{i \in \mathcal{I}}$ a joint action set, and $\big(\mathbb{J}_i\big)_{i \in \mathcal{I}}$ a collection of utility functions $\mathbb{J}_i \colon \mathcal{U} \to \mathbb{R}$. A game is called *potential* if there exists a function $\mathbb{J} \colon \mathcal{U} \to \mathbb{R}$, such that the following equality holds

$$\mathbb{J}_i(\boldsymbol{u}_i', \boldsymbol{u}_{-i}) - \mathbb{J}_i(\boldsymbol{u}_i, \boldsymbol{u}_{-i}) = \mathbb{J}(\boldsymbol{u}_i', \boldsymbol{u}_{-i}) - \mathbb{J}(\boldsymbol{u}_i, \boldsymbol{u}_{-i}),$$

where $\big(\boldsymbol{u}_i,\, \boldsymbol{u}_{-i}\big)$, $\big(\boldsymbol{u}_i',\, \boldsymbol{u}_{-i}\big) \in \mathcal{U}$ are two action profiles that are identical up to agent i.
The following holds for the global objective (4.7):

$$\begin{aligned}
\mathbb{J}\big((\boldsymbol{u}_i', \boldsymbol{u}_{-i}), \mathfrak{R}(\boldsymbol{u}_i', \boldsymbol{u}_{-i})\big) - &\mathbb{J}\big((\boldsymbol{u}_i, \boldsymbol{u}_{-i}), \mathfrak{R}(\boldsymbol{u}_i, \boldsymbol{u}_{-i})\big) = \\
&= \int_{\mathcal{R}_i(\boldsymbol{u}_i', \boldsymbol{u}_{-i})} \|\boldsymbol{u}_i' - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q} + \sum_{j \in \mathcal{I} \setminus i} \int_{\mathcal{R}_j(\boldsymbol{u}_i', \boldsymbol{u}_{-i})} \|\boldsymbol{u}_j - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q} \ \ldots \\
&\quad - \int_{\mathcal{R}_i(\boldsymbol{u}_i, \boldsymbol{u}_{-i})} \|\boldsymbol{u}_i - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q} - \sum_{j \in \mathcal{I} \setminus i} \int_{\mathcal{R}_j(\boldsymbol{u}_i, \boldsymbol{u}_{-i})} \|\boldsymbol{u}_j - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q} \\
&= \int_{\mathcal{R}_i(\boldsymbol{u}_i', \boldsymbol{u}_{-i})} \|\boldsymbol{u}_i' - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q} + \sum_{j \in \mathcal{N}_i} \int_{\mathcal{R}_j(\boldsymbol{u}_i', \boldsymbol{u}_{-i})} \|\boldsymbol{u}_j - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q} \ \ldots \\
&\quad - \int_{\mathcal{R}_i(\boldsymbol{u}_i, \boldsymbol{u}_{-i})} \|\boldsymbol{u}_i - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q} - \sum_{j \in \mathcal{N}_i} \int_{\mathcal{R}_j(\boldsymbol{u}_i, \boldsymbol{u}_{-i})} \|\boldsymbol{u}_j - \boldsymbol{q}\|^2 \varrho(\boldsymbol{q}) d\boldsymbol{q} \\
&= \mathbb{J}_i((\boldsymbol{u}_i', \boldsymbol{u}_{-i}), \mathfrak{R}(\boldsymbol{u}_i', \boldsymbol{u}_{-i})) - \mathbb{J}_i((\boldsymbol{u}_i, \boldsymbol{u}_{-i}), \mathfrak{R}(\boldsymbol{u}_i, \boldsymbol{u}_{-i}))
\end{aligned}$$

where the structure of $\mathbb{J}_i$ is the same of (4.9).