Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Automatic Control Laboratory

Semester Project

Distributed Robotic Swarms Coordination via Feedback Equilibrium Seeking

Andrea Da Col
January 2022

Advisors
Dr. Giuseppe Belgioioso, Prof. Florian Dörfler

Acknowledgements

I would like to express my sincere gratitude to my primary supervisor, Dr. Giuseppe Belgioioso, for the support, patience, and dedicated involvement he has shown throughout the duration of this project. Your insightful feedback and expertise were really influential in sharpening my interest and critique sense towards subjects that were new to me.

Abstract

Robotic formation control has been a topic of great interest in recent times due to the new possibilities and real-world applications it has disclosed, especially in wireless communication networks. In this context, *optimal relay placement* refers to the problem of coordinating a network of robotic relays in order to accomplish an arbitrary global objective.
In this project, we employ a game-theoretic approach for steering robotic relays, both terrestrial (differential drives) and aerial (quadrotors) to a desired a-priori unknown configuration in a fully-distributed and autonomous fashion. To account for the presence of external perturbations, such as aerodynamic forces, that may prevent the system from converging to the goal configuration, appropriate local controllers are designed to attain closed-loop stability, disturbance rejection, and reference tracking. An high-level controller based on the recently proposed Feedback Equilibrium Seeking (FES) control paradigm, plays the role of a trajectory planner. Selected simulations, involving two and three-dimensional formations of differential drives and quadrotors respectively, both with non-linear dynamics, are presented to corroborate the theory.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Consider the problem of two ground users exchanging data while moving in a known environment. In general, the information transfer is not always possible due to large distances, shadowing, and multipath fading. Reliable end-to-end communication links can thus be obtained with the help of intermediate agents that cooperatively place themselves between the transmitter (TX) and receiver (RX) to meet a global objective. On-demand and quick deployment, high mobility in harsh environments, make robotic relays suitable for a variety of civilian applications, including search and rescue, inspection, wireless communications. Traffic offloading, establishment of wireless links between distant users without direct connection, service recovery in emergency situations are only some of the examples explored in the literature [1], [2], [3].

The success of robotic operations in wireless networks strongly depends on the ability of such group of agents to maintain a certain degree of connectivity throughout time. One agent must choose carefully which action to play in order to preserve existing connections to its neighbours. Communication-aware path planning is fundamental in the formation control of a wireless relay network, as the quality of the channel may undergo fast spatial variations [4].

1.1 Literature Review

The literature on optimal relay placement is vast and heterogeneous. Lots of metrics have been proposed as a definition for optimal deployment of a robotic relay network, depending on the real-world application. In this section, we list the main results on optimal terrestrial and aerial robotic relay deployment in wireless networks.

Methods based on approximate solutions to distributed optimization problems, as well as game theoretic approaches for multi-robot path planning are presented in [5]. Interference and capacity saturation of the relay's channel were considered in [6] to achieve maximum-capacity deployment in cellular networks. [7] extends to the case of optimization based on the distribution of the users in non-ideal physical environments. [8] presents joint optimization of communication and motion for a multi-robot system in order to establish a stable connection between users in a fading environment, under resource availability constraints.

The need for high mobility in challenging environments and the ability to establish reliable communication links with ground users due to an high probability of line-of-sight (LoS) has moved the attention towards UAV relays. The single UAV scenario [9] has been widely discussed in the literature, but it fails when it comes enable long-distance communications in complicated environments, thus motivating the use of a network of UAVs. Algorithms for distributed path planning and end-to-end throughput maximization via transmit power optimization are studied in [1] and [3], [10] respectively. Motivated by the short lifetime of the state-of-the-art batteries, [11] has proposed tethered UAVs (T-UAVs) as an alternative to the regular ones in long-term

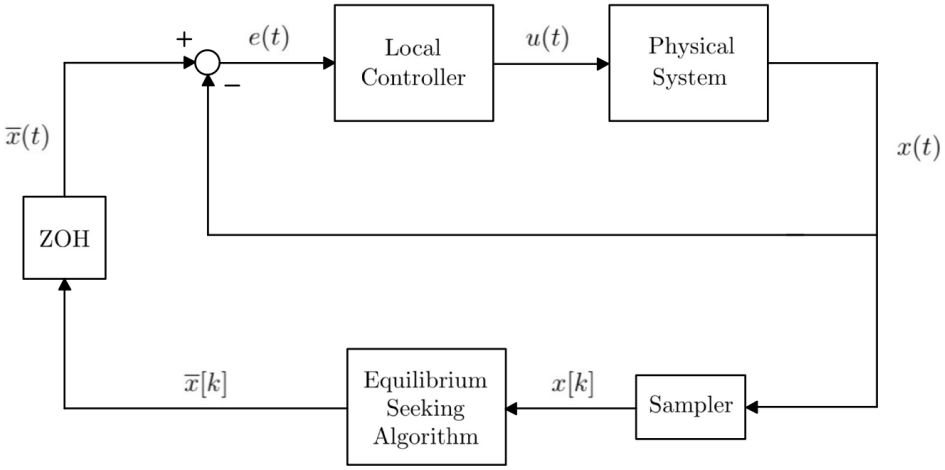

Figure 1.1: Block diagram of the proposed control architecture.

missions, at the expense of a reduced freedom of mobility. Their performance is compared in a heavy traffic scenario.

The presence of external, possibly not measurable disturbances, as well as the need for large networks of potentially self-interested robotic relays, constitute some of the main limitations for the majority of the works that have have been listed before, which restrict to small multi-agent systems operating in unperturbed environments, thus lacking scalability and robustness. With this in mind, and motivated by some recent advances in game-theoretic robotic networks formation [5] and equilibrium seeking algorithms, we propose a control strategy for the problem of optimal relay placement with moving ground stations, in the case of realistic wireless communications and unmeasured external disturbances.

1.2 Contributions

In this project, we aim at tailoring the FES control paradigm for robotic swarms coordination. The FES framework is suitable for the optimal relay placement problem since it allows to model a variety of useful objectives and to express multi-agent communication as a discrete process using well-developed distributed optimization algorithms. Further, the network systems can be driven to efficient equilibrium points obtained as the solutions to a generalized equation, thus providing appropriate tools for encoding competition and cooperation [12]. The use of a distributed approach is justified by intrinsic limitations of wireless settings: one relay shares links only with those that are close to it. Our contributions are twofold:

1. We design a distributed, scalable, robust control strategy for optimal multi-relay placement in wireless networks to maximize the probability of correct end-to-end signal propagation.

2. We provide simulations that show convergence to efficient equilibrium points in the presence of environmental disturbances both in the case of terrestrial and aerial relay networks.

1.3 Report Structure

The remainder of this report is organized as follows. Chapter 2 presents the continuous-time dynamics for two types of autonomous mobile relays, namely, differential drives and quadrotors.

In our discussion, optimal placement of a planar formation of terrestrial relays is used as a starting point for a generalization to the case of three-dimensional robotic coordination. Local control architectures providing stability guarantees are developed in Chapter 3. The design of an high-level path planner (or guidance controller) is addresses in Chapter 4, where a game formulation for the coordinated control of the multi-agent system and a distributed discrete algorithmic approach to the solution are discussed. In Chapter 5 we present a FES-based feedback controller that drives the continuous-time plant to the a-priori unknown optimal network configuration. Chapter 6 provides simulation results to validate the performance of the proposed control paradigm and Chapter 7 concludes the report.

1.4 Preliminaries

A good, or at least easier, understanding of the proposed framework for the optimal relay placement problem presumes the reader to have solid basis in mobile-base robot dynamics, control theory, and communication networks. Throughout our discussion we motivate and justify our assumptions by means of some important results attained in such well-studied topics. Nonetheless, the reader must be aware that this work is not meant to be a detailed presentation of such achievements. To this end, in what follows, we suggest a list of references for the inexperienced reader in order to make clear the interplay between technical notions of robotics, control systems, wireless communications. For a detailed description on modelling and control of wheeled and aerial robotic relays we refer to [13], [14], [15], [16]. Other introductory notions of robotics can be found in [17]. [18] provides a self-contained account of the main results in convex optimization and monotone operator theory, which are extensively used in Game Theory. A rigorous design framework, as well as a closed-loop stability analysis for Feedback Equilibrium Seeking in nonlinear systems, are derived in [12] as an extension to [19]. Finally, for an accepted overview on wireless communications in realistic environments, and channel quality-dependent path planning of autonomous wheeled robots the reader is referred to [4], and [8], [20], respectively.

Chapter 2

Relay Modelling

In this chapter, we present the equations of motion (EoMs) that model the dynamics of the relays. We start by considering terrestrial robotic relays and then move to the more involved case of unmanned aerial vehicles.

2.1 Terrestrial Robot Dynamics

We consider a swarm of n differential drives moving on the cartesian plane and that evolve according to the following unicycle dynamics:

$$\begin{aligned} \dot{a}_i &= v_i \cos\theta_i \\ \dot{b}_i &= v_i \sin\theta_i \\ \dot{\theta}_i &= \omega_i \end{aligned} \tag{2.1}$$

where a_i and b_i are the position coordinates expressed with respect to a fixed cartesian reference frame, θ_i is the steering angle, v_i and p_i are the linear and angular velocity of the i-th agent. Let us collect them in the state vector $\mathbf{x}_i = [a_i \;\; b_i \;\; \theta_i]^\top$ and the control input vector $\mathbf{u}_i = [v_i \;\; \omega_i]^\top$. The EoMs in (2.1) constitute a clear example of *maximally non-holonomic* system: motion cannot happen immediately in an arbitrary direction but is constrained to be orthogonal to the direction of the wheel axis at any given time. This behaviour can be expressed through the following non-holonomic constraint on the velocity:

$$\dot{a}_i \sin\theta_i - \dot{b}_i \cos\theta_i = 0. \tag{2.2}$$

Hence, the system's state $\mathbf{x}_i(t)$ is not allowed to describe any possible trajectory in the state space, but only those for which (2.2) is met at each time. However, it can be shown that there exists a control input sequence driving the system from its initial state to any desired configuration. In this sense, *maximal non-holonomy* is equivalent to controllability.

2.1.1 External Disturbances

Eqn. (2.1) describes the state evolution in ideal conditions, thus not accounting for unmeasured exogenous disturbances that may make the robot deviate from the desired trajectory. Main perturbations that affect a relay can be found in aerodynamic forces such as wind gusts, irregular terrain conditions, side-slip due to non-ideal constraints in (2.2). Perturbations of this kind are typically modelled as an additive disturbance $\mathbf{w} = [w_{x,i} \;\; w_{y,i}]^\top \in \mathcal{W}$ affecting both the horizontal

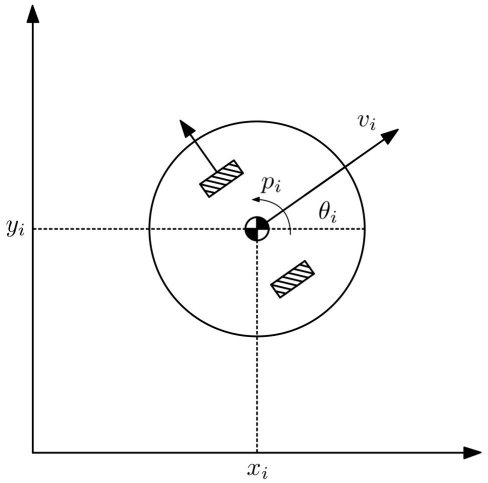

Figure 2.1: Non-holonomic relay with unicycle dynamics

and vertical motion, where $\mathcal{W} \subset \mathbb{R}^2$ is a bounded set. In our setting, we consider the case of exogenous, unmeasurable disturbances of the form:

$$\begin{aligned} \dot{a}_i &= v_i \cos\theta_i + w_{a,i} \\ \dot{b}_i &= v_i \sin\theta_i + w_{b,i} \\ \dot{\theta}_i &= \omega_i. \end{aligned} \tag{2.3}$$

2.2 Quadrotor Robot Dynamics

In this section, the focus is put on quadrotors, also called quadcopters, as examples of UAVs. The ability to hover is place, vertically take off and land makes them preferable to fixed-wings UAVs in the context of *optimal relay placement*, which presumes the agents to keep an optimal position once it is reached.

From now on we will make extensive use of the concepts of inertial frame, body frame, and coordinates transformations. Readers that are not familiar with these notions are referred to [17]. Both the mathematical model and the notation are inspired by [16], in which non-linear differential equations for the dynamics of a quadcopter are derived using the Newton-Euler approach. Before addressing the ODEs' structure, we provide a brief explanation of the main parameters and states variables. Once again, consider a swarm of n identical aerial robotic relays, each governed by the non-linear dynamics:

$$\boldsymbol{\xi}_i = \begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix}, \qquad \boldsymbol{\eta}_i = \begin{bmatrix} \phi_i \\ \theta_i \\ \psi_i \end{bmatrix}, \qquad \mathbf{v}_i^B = \begin{bmatrix} v_{x,i} \\ v_{y,i} \\ v_{z,i} \end{bmatrix}, \qquad \boldsymbol{\nu}_i = \begin{bmatrix} p_i \\ q_i \\ r_i \end{bmatrix} \tag{2.4}$$

where $\boldsymbol{\xi}_i$ contains the coordinates of the quadcopter's center of gravity (CoG) expressed in the inertial frame [**x** **y** **z**], $\boldsymbol{\eta}_i$ defines the attitude as a set of roll, pitch, and yaw angles. Linear velocities $\mathbf{v}_i^B$ and angular rates $\boldsymbol{\nu}_i$ are written with respect to the body-fixed frame coordinates

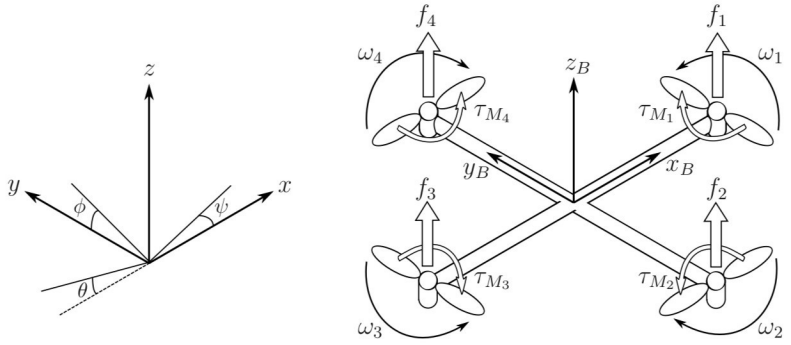

Figure 2.2: Inertial and body-fixed frames of the quadcopter [16].

$[\mathbf{x}^B \ \mathbf{y}^B \ \mathbf{z}^B]$. Let us define the stacked vector:

$$\mathbf{x}_i = \begin{bmatrix} \boldsymbol{\xi}_i \\ \boldsymbol{\eta}_i \\ \mathbf{v}_i^B \\ \boldsymbol{\nu}_i \end{bmatrix}, \tag{2.5}$$

which is the 12-dimensional state of the i-th agent.

Under the assumption of propellers placed at equal distance l from the CoG of the main rigid body, which implies symmetry in the xz-plane and yz-plane, the inertia matrix $\boldsymbol{I}$ has diagonal structure, i.e.,

$$\boldsymbol{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}.$$

Linear and angular motion of the main body are obtained through the rotation of the propellers. The angular velocity $\omega_{k,i}$ of a rotor $k \in \{1, 2, 3, 4\}$ of the i-th agent generates a force $F_{k,i}$ pointing upwards, i.e., in the direction of $\mathbf{z}^B$, and a torque $\tau_{M_{k,i}}$ about its axis of rotation. Based on this, it is possible to define auxiliary control inputs to the i-th UAV:

$$\boldsymbol{T}_i^B = \begin{bmatrix} 0 \\ 0 \\ T_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sum_{k=1}^4 F_{k,i} \end{bmatrix} \qquad \boldsymbol{\tau}_i^B = \begin{bmatrix} \tau_{\phi,i} \\ \tau_{\theta,i} \\ \tau_{\psi,i} \end{bmatrix} = \begin{bmatrix} lk(-\omega_{2,i}^2 + \omega_{4,i}^2) \\ lk(-\omega_{1,i}^2 + \omega_{3,i}^2) \\ \sum_{k=1}^4 \tau_{M_{k,i}} \end{bmatrix} \tag{2.6}$$

given by the combined action of the propellers. Eqn. (2.6) indeed shows that a total vertical thrust $\boldsymbol{T}_i^B$, as well as roll, pitch, and yaw moments $\boldsymbol{\tau}_i^B$ are generated by regulating the difference in squared angular velocity between the rotors, thus resulting in a vertical, lateral and angular motion of the main rigid body. Thus, the definition of auxiliary control inputs allows a much easier understanding of global the system's dynamics in terms of total forces and torques acting on the quadcopter. With this in mind, we can write the non-linear dynamics of the uncontrolled system using the Newton–Euler method for the linear and the angular motion [16]:

$$\begin{gathered} m\ddot{\boldsymbol{\xi}}_i = \boldsymbol{G} + \boldsymbol{R}_i \boldsymbol{T}_i^B \\ \begin{bmatrix} \ddot{a}_i \\ \ddot{b}_i \\ \ddot{c}_i \end{bmatrix} = -g \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + \frac{T_i}{m} \begin{bmatrix} \cos\psi_i \sin\theta_i \cos\phi_i + \sin\psi_i \sin\phi_i \\ \sin\psi_i \sin\theta_i \cos\phi_i - \cos\psi_i \sin\phi_i \\ \cos\theta_i \cos\phi_i \end{bmatrix} \end{gathered} \tag{2.7}$$

$$
\begin{gathered}
\boldsymbol{I}\dot{\boldsymbol{\nu}}_i + \boldsymbol{\nu}_i \times (\boldsymbol{I}\boldsymbol{\nu}_i) + \boldsymbol{\Gamma}_i = \boldsymbol{\tau}_i^B \\
\begin{bmatrix} \dot{p}_i \\ \dot{q}_i \\ \dot{r}_i \end{bmatrix} = \begin{bmatrix} q_i r_i (I_{yy} - I_{zz})/I_{xx} \\ p_i r_i (I_{zz} - I_{xx})/I_{yy} \\ p_i q_i (I_{xx} - I_{yy})/I_{zz} \end{bmatrix} + \begin{bmatrix} \tau_{\phi,i}/I_{xx} \\ \tau_{\theta,i}/I_{yy} \\ \tau_{\psi,i}/I_{zz} \end{bmatrix} + \boldsymbol{\Gamma}_i
\end{gathered} \tag{2.8}
$$

where $\boldsymbol{G}$ is the gravity vector in the inertial frame, $\boldsymbol{R}_i$ is the rotation matrix from the body frame to the inertial frame, retrieved from $\boldsymbol{\eta}_i$, and $\boldsymbol{\Gamma}_i$ is a term that accounts for gyroscopic effects. Aerodynamic effects, including hub forces arising in the horizontal motion, as well as inertial counter torques and rolling moments, shall also be considered in the modelling of a quadrotor, as a generalization to (2.7) and (2.8). Equations accounting for drag forces can be found in [16]. Despite being difficult to express mathematically, such non-ideal behaviours can be neglected when dealing with small attitude angles.

Chapter 3

Local Controller Design

Before addressing the problem of distributed formation control, it is important to ensure certain properties in the local control of each agent. A robotic relay should describe a trajectory that is close to that computed by the upper-level guidance controller, hence the need for stability and good tracking performances. In this chapter, we discuss local control architectures with guarantees of stability and reference tracking performance both in the planar and the three-dimensional case.

3.1 Non-linear Controller for a Relay with Unicycle Dynamics

In the previous chapter, we introduced unicycle robots as examples of controllable systems featuring maximal non-holonomy. We are concerned with controlling the state $\mathbf{x}_i \in \mathcal{X}$, $\mathcal{X} \subseteq \mathbb{R}^3$ using control signals $\mathbf{u}_i \in \mathcal{U}$, where $\mathcal{U} \subseteq \mathbb{R}^2$. Clearly, only certain control directions are allowed for a given configuration of the relays, as we can see by rewriting eq. (2.1) as the following driftless affine-control system:

$$\begin{bmatrix} \dot{a}_i \\ \dot{b}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \\ 0 \end{bmatrix} v_i + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_i. \tag{3.1}$$

Controllability of the robotic system is meant in the sense of *small-time local controllability* (STLC) and can be assessed using the Lie algebra rank conditions on its involutive closure. By evaluating the Lie bracket $[f, g]$ of the vector fields $f = [\cos\theta_i \;\; \sin\theta_i \;\; 0]^\top$ and $g = [0 \;\; 0 \;\; 1]^\top$:

$$[f, g]_j = \sum_{k=1}^{3} \left(f_k \frac{\partial g_j}{\partial x_k} - g_k \frac{\partial f_j}{\partial x_k} \right), \qquad j = \{1, 2, 3\}, \tag{3.2}$$

and leveraging the well-known results of the Chow–Rashevskii theorem [14] for driftless affine-control systems, it is easy to check that the *Lie algebra* $\mathcal{L}(\Delta)$ of a differential drive is three-dimensional. For a complete overview on STLC of underactuated systems, we refer to [14], [13].

Consider the single relay in the network, tasked with the tracking of position commands computed by the upper-level path planner. The robotic agent will implement a globally asymptotically stable controller, designed using backstepping [15], that performs smooth tracking of a time varying set-point. Further, we assume that it has access to the measurements y_i in relative position from their neighbours $j \in \mathcal{N}_i$. The (local) control policy is:

$$\begin{aligned} v_i &= k_1 \|\mathbf{r}_i - \overline{\mathbf{r}}_i\|_2 \cos\phi_i \\ \omega_i &= -k_1 \cos\phi_i \sin\phi_i - k_2 \phi_i \end{aligned} \tag{3.3}$$

where $\mathbf{r}_i = [a_i \;\; b_i]$ and $\bar{\mathbf{r}}_i = [\bar{a}_i \;\; \bar{b}_i] \in \mathbb{R}^2$ are, respectively, the actual position and the position command in xy-coordinates, $\phi_i = \pi + \theta_i - \arctan\left(\frac{b_i - \bar{b}_i}{a_i - \bar{a}_i}\right)$ is the heading error, k_1 and k_2 are control gains. Closed-loop asymptotic stability can be proved by constructing an appropriate Lyapunov function [15], i.e.,

$$V(\mathbf{x}) = \frac{1}{2}\left(\|\mathbf{r}_i - \bar{\mathbf{r}}_i\|_2^2 + \phi_i^2\right) \tag{3.4}$$

that depends on the linear distance to the target and the heading errors.

3.2 Control of a Quadrotor

Underactuated systems with six degrees of freedom and four independent control inputs, quadcopters are difficult to control due to their highly non-linear dynamics. Solutions employing a Linear Quadratic Regulator (LQR) and regular PID synthesis have been extensively studied in the literature [21], [22]. However, they fail to capture non-linearity in the dynamics of the plant, resulting in poor control policies that are meaningful only when the system works around a desired operating point. An LQR-based approach would indeed provide an optimal linear controller for the stabilizable linearized model, but struggles when it comes to constrained state and action spaces. More sophisticated methods achieve stabilization of the translational and rotational motion using non-linear PID controllers for energy-efficient path tracking [23], or receiding-horizon techniques [24]. In the next section, we will design a Model Predictive Control (MPC) controller to account for the non-linear EoMs in (2.7)-(2.8) and for input saturation.

3.2.1 Non-Linear MPC

We assume that each UAV is implementing a (local) non-linear MPC. To make notation simpler, let us drop the subscript i when indicating the i-th agent in the formation.

MPC solves a finite-horizon optimal control problem online, using the current state as initial condition, to compute the next control input. In our setup we assume perfect knowledge of such state. At each sampling period, an open-loop control sequence of length N (if one exists) is returned as a solution to a constrained optimization problem, and only the first element of it is applied to the system. The procedure is then repeated in a receiding-horizon fashion, thus achieving some form of state feedback. The general formulation of a non-linear MPC optimal control problem is

$$\begin{aligned}
\min_{U} \quad & l_f(\mathbf{x}_N) + \sum_{j=0}^{N-1} l(\mathbf{x}_j, \mathbf{u}_j) \\
\text{s.t.} \quad & \mathbf{x}_{j+1} = f(\mathbf{x}_j, \mathbf{u}_j), && \forall j \in \{0, \dots, N-1\} \\
& \mathbf{x}_j \in \mathcal{X}, && \forall j \in \{0, \dots, N-1\} \\
& \mathbf{u}_j \in \mathcal{U}, && \forall j \in \{0, \dots, N-1\} \\
& \mathbf{x}_N \in \mathcal{X}_f \\
& \mathbf{x}_0 = \mathbf{x}(k)
\end{aligned} \tag{3.5}$$

. Recursive feasibility and stability guarantees are generally obtained designing an appropriate terminal set $\mathcal{X}_f$ which is invariant under a local control law, and the terminal cost $l_f : \mathcal{X}_f \to \mathbb{R}_{\geq 0}$ as a Lyapunov function for the non-linear system. The rationale is the following: we are forcing the terminal state $\mathbf{x}_N$ of the open-loop state evolution to fall in a terminal set induced by a stabilizing controller because we know that, once there, the controlled system will converge to the desired reference asymptotically, collecting an infinite-horizon cost l_f that can be computed in closed form. However, it is difficult to find invariant sets and Lyapunov functions for non-linear

systems in general.

We are almost ready to give a formulation for the MPC tracking problem. Consider the reference trajectory from an upper-level path planner as a piecewise constant signal on time intervals with the same length. At each sampling period, the local non-linear MPC will track the aforementioned output reference, which is constant in the time interval under consideration:

$$\begin{aligned}
\min_{U} \quad & \sum_{j=0}^{N-1} \|\mathbf{x}_j - \overline{\mathbf{x}}\|_Q + \|\mathbf{u}_j - \overline{\mathbf{u}}\|_R + \|\mathbf{u}_j - \mathbf{u}_{j-1}\|_W \\
\text{s.t.} \quad & \mathbf{x}_{j+1} = f(\mathbf{x}_j, \mathbf{u}_j), && \forall j \in \{0, \dots, N-1\} \\
& \mathbf{x}_j \in \mathcal{X}, && \forall j \in \{0, \dots, N-1\} \\
& \mathbf{u}_j \in \mathcal{U}, && \forall j \in \{0, \dots, N-1\} \\
& \mathbf{x}_N = \overline{\mathbf{x}} \\
& \mathbf{x}_0 = \mathbf{x}(k)
\end{aligned} \tag{3.6}$$

where we require the state trajectory to reach the steady-state target $\overline{\mathbf{x}}$ at the end of the prediction horizon. Notice that this is the simplest choice for a positive invariant terminal set since it does not require a terminal cost to be designed. The downside is that feasibility of a trajectory may now strongly depend on the length of the prediction horizon. The control input to keep this equilibrium, once obtained, is denoted with $\overline{\mathbf{u}}$.

In eq. (3.6) $P \succeq 0$ and $R \succ 0$, are weight matrices that encode reference tracking performance for the state $\mathbf{x}$ and the manipulated variable $\mathbf{u}$, respectively. $W \succ 0$ is a penalty that discourages fast changes in the control action. In particular, we focus on diagonal matrices:

$$Q = \begin{bmatrix} q_1 & & \\ & \ddots & \\ & & q_{12} \end{bmatrix}, \quad R = \begin{bmatrix} r_1 & & \\ & \ddots & \\ & & r_4 \end{bmatrix}, \quad W = \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_4 \end{bmatrix} \tag{3.7}$$

and take inspiration from [24] to set the weights. Motivated by the fact that good positioning in wireless networks is critical, we choose to penalize large differences from the target state $\overline{\mathbf{x}}$ more than we do for control commands that are far from the steady state action $\overline{\mathbf{u}}$.

Chapter 4

Coordination Game and Algorithms

Motivated by the limitations arising in wireless communications, we recognize the distributed, game-theoretic framework as a better tool to encode useful objectives in multi-robot systems potentially including self-interested agents.

In this chapter, we are concerned with the formulation of a (generalized) cooperative potential game to model a desired configuration of the network of relays where the agents' locations are expressed as generalized Nash equilibria (GNEs).

4.1 Connectivity Constraints via Generalized Potential Games

The main challenge in cooperative control is to assign each player an appropriate local utility function such that a global network objective is accomplished. Inspired by [25], where examples of utility functions for multi-robot applications are presented in a learning-in-games framework, we formulate a generalized potential game to model the following network objective: maximize the probability of correct end-to-end signal propagation in a realistic wireless communications setting [8]. Generalized potential games are indeed suitable for us, as they simplify design of such local utility functions, while allowing to model agents (or players) with local interests.

Let us define the set $\mathcal{I} := \{1, \dots, n\}$ of all players (i.e., the relays) and the set of all the m transmitting and receiving stations $\mathcal{J} := \{1, \dots, m\}$. To model bidirectional exchange of data in communication networks, we adopt a time-invariant undirected graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \mathcal{I} \cup \mathcal{J}$ is the set of nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a collection of unordered pairs of nodes $\{i, j\}$, called edges, that model enabled wireless connection between agents i and j. In addition, consider $\mathcal{G}$ to be connected, i.e., there exists a path connecting any two nodes. This last assumption is needed for the optimal relay placement problem to be well-defined.

Consider the action space $\Omega_i \subset \mathbb{R}^{n_x}$ as a set of feasible spatial configurations for the i-th player, and its local utility function $U_i : (x_i, x_{-i}) \mapsto U_i(x_i, x_{-i})$, where $\Omega := \prod_{i \in \mathcal{I}} \Omega_i$, to express optimality of a configuration. In our case, Ω_i is compact and convex, for all players $i \in \mathcal{I}$.
We use the following definition of a potential game, taken from [25]:

Definition 4.1.1 (Potential Games)**.** A set of action spaces $\{\Omega_i\}_{i=1}^n$ and and relative utilities $\{U_i\}_{i=1}^n$ constitutes a potential game if there exists a potential function $\phi : \Omega \to \mathbb{R}$ such that:

$$U_i(x_i'', x_{-i}) - U_i(x_i', x_{-i}) = \phi(x_i'', x_{-i}) - \phi(x_i', x_{-i}) \tag{4.1}$$

for all $i \in \mathcal{I}$, for all x_i', $x_i'' \in (\Omega_i)$ and for all $x_{-i} \in \Omega \setminus \Omega_i$.

In the next sections, we exploit the results in [4] and [8] to design local utilities for the 2D and 3D optimal relay placement problems.

4.1.1 Utility Design for Unicycle Players

The global objective consists in propagating a signal that contains no flipped bits to a receiving node (RX). Given the stochastic behaviour of communication links between two communicating agents, due to channel fading and shadowing effects that may arise in realistic environments, the network objective is formulated in terms of probability of correct reception.

Consider now an arbitrary connected graph structure and a number m of transmitting (TX) and receiving (RX) base stations in $\mathcal{J}$, $m \geq 2$. Call $\mathcal{P}$ the path connecting two distinct elements $\mathrm{TX}, \mathrm{RX} \in \mathcal{J}$ (with no cycles), then define the SNR in the transmission from two consecutive nodes, e.g., $i-1, i \in \mathcal{P}$:

$$\Upsilon_{\mathrm{SNR,PL}}(x_{i-1}, x_i) = \frac{\alpha_{\mathrm{SNR,PL}}}{\|x_{i-1} - x_i\|^{n_{\mathrm{PL}}}} \tag{4.2}$$

where $x_{i-1}, x_i \in \mathbb{R}^2$ are vectors containing the xy-coordinates of the two nodes, while $\alpha_{\mathrm{SNR,PL}}$ and n_{PL} are path loss parameters that encode quality of the signal as a function of the distance between them. If the probability of correct reception at node RX, denoted as $\mathbb{P}(\mathrm{RX})$, is expressed in terms of the probability of correct reception at the previous nodes in the path $\mathcal{P}$, then we can leverage the results in [8] and write the following expression for the probability of correct signal propagation along the fixed path $\mathcal{P}$:

$$\mathbb{P}(\mathrm{RX}) = \prod_{i=2}^{|\mathcal{P}|} \left(1 - 0.2 \exp\left(-c\Upsilon_{\mathrm{SNR,PL}}(x_{i-1}, x_i)\right)\right) \tag{4.3}$$

where $c \in \mathbb{R}_{\geq 0}$ is a non-negative constant that depends on the size of the relay network. By taking the logarithm of (4.3), which preserves monotonicity, we obtain a more tractable version of the global utility function, which is strongly concave on Ω if $\alpha_{\mathrm{SNR,PL}}$ and n_{PL} are selected appropriately:

$$\phi = \sum_{i=2}^{|\mathcal{P}|} \ln\left(1 - 0.2 \exp\left(-c\Upsilon_{\mathrm{SNR,PL}}(x_{i-1}, x_i)\right)\right) \tag{4.4}$$

Notice that, in our description, we are assuming that the players have perfect knowledge of the channel parameters $\alpha_{\mathrm{SNR,PL}}$ and n_{PL}, which express the transmitting power of the wireless antenna mounted by each relay, and the power fall-off rate, respectively. However, in most cases the channel is not deterministic and the parameters have to be estimated after collecting several measurements [4]. Furthermore, we neglect phenomena like shadowing and multipath fading, which are highly dependent on the configuration of the environment, and choose to focus only on distance-dependent path loss effects.

The local utility assigned to each player $i \in \mathcal{I} \cap \mathcal{P}$ is defined as

$$U_i(x_i, x_{-i}) = \sum_{j \in \mathcal{N}_i \cap \mathcal{P}} \ln\left(1 - 0.2 \exp\left(-c\Upsilon_{\mathrm{SNR,PL}}(x_i, x_j)\right)\right). \tag{4.5}$$

It can be shown that (4.4) and (4.5) satisfy the conditions in Definition 4.1.1. Therefore, we then have the following potential game:

$$\begin{aligned} \forall i \in \mathcal{I}: \quad & \min_{x_i} \quad -U_i(x_i, x_{-i}) \\ & \text{s.t.} \quad x_i \in \Omega_i \end{aligned} \tag{4.6}$$

The solution of the game in (4.6), i.e., the Nash equilibria, define the shape of the optimal path connecting TX and RX. In particular, the relays will be equally spaced on the segment connecting the extrema of $\mathcal{P}$, if they share the same distance-dependent path loss parameters (i.e., $\alpha_{\mathrm{SNR,PL}}$ and n_{PL}) and the connectivity constraints are met.

The reader is invited to note that the proposed setting can be easily extended to an arbitrary number of paths linking pairs of base stations and that, in general, each player can be part of multiple paths, thus resulting in more complicated optimal network configurations. Once again, we recall that the network topology, as well as all the links connecting two moving base stations are chosen a-priori and do not vary during the control of the robotic swarm: online rearrangement of the aforementioned communication paths would make the problem much more difficult to investigate, as non-linearities may arise when different feasible sequences of intermediate nodes, connecting a given pair TX and RX, are selected. Here, feasibility of an ordered sequence of nodes is meant with respect to the assumption of finite communication range.

4.1.2 Utility Design for UAV Players

The ability to move in complicated environments, as well as improved line-of-sight (LoS) are the main reasons to consider networks of UAVs. Obstacles, that are responsible for shadowing and fading effects in the planar case, do not play a major role in terms of quality of the communication channel, if we allow three-dimensional motion. Whit this in mind, we propose a second, simpler global utility function, which does not make use of the path loss parameters, i.e., θ_{PL}.

Consider now a much simpler scenario than that in Section 4.1.1, namely a next-hop graph topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ linking two ground stations. Recalling that the players should lie on the straight line connecting the transmitting and receiving nodes, we propose the following global objective:

$$\phi = \sum_{\{i,j\}\in\mathcal{E}} \|x_i - x_j\|_2^2 \tag{4.7}$$

that we aim to minimize. Notice that now $x_i, x_j \in \mathbb{R}^3$ indicate the three-dimensional positions of the CoG of UAV i and j, respectively. Local utilities are easily derived from (4.7):

$$U_i(x_i, x_{-i}) = \sum_{j\,\in\,\mathcal{N}_i} \|x_i - x_j\|_2^2 \tag{4.8}$$

Likewise in the unicycle example, we are left with a collection of n interdependent optimization problems that define the potential game

$$\begin{aligned} \forall i \in \mathcal{I}: \quad & \min_{x_i} \quad U_i(x_i, x_{-i}) \\ & \text{s.t.} \quad x_i \in \Omega_i \end{aligned} \tag{4.9}$$

4.1.3 Coupling Constraints

Connectivity of the network structure throughout time is critical in a wireless setting. Take any player $i \in \mathcal{I}$ that it is also an element of a path $\mathcal{P}$ connecting to ground users. This player has two tasks to accomplish: minimize the total squared euclidean distance from its neighbouring nodes along $\mathcal{P}$ while staying sufficiently close to those in $\mathcal{N}_i$. In general, convergence to the optimal configuration is undesirable if one agent cannot stay within the finite wireless communication range of all its neighbours during this process. By introducing coupling among the decision sets, one player should choose the next move in a more conservative way, thus prioritizing cohesion of the underlying graph structure over fast convergence to the optimal formation shape.

We decide to enforce coupling constraints using the affine function $x \to Ax - b$, where $A := [A_1|\dots|A_n] \in \mathbb{R}^{|\mathcal{E}|\times n_x n}$, $x^\top := [x_1^\top \dots x_n^\top]$ and $b := \sum_{i=1}^n b_i \in \mathbb{R}^{|\mathcal{E}|}$, as in [26]. In particular, we impose all pairs of neighbouring nodes in the graph to stay sufficiently close in the sense of $\|\cdot\|_\infty$:

$$\|x_i - x_j\|_\infty \le R_{\max}, \quad \forall i \in \mathcal{I}, \quad \forall j \in \mathcal{N}_i \tag{4.10}$$

where $R_{\max} \in \mathbb{R}_{>0}$ is a scalar denoting the maximum length an edge $\{i,j\} \in \mathcal{E}$ can take, for all $i,j \in \mathcal{V}$. Notice that the equations in (4.10) can be more compactly written in terms of A, x and b, and that the choice of the infinity-norm ball can be extended to an arbitrary convex polyedron. Coupling constraints have the effect of shrinking the set of actions one player is allowed to take: a relay will not move in a certain direction if this means cutting a communication link with a node in its vicinities. The feasible action set associated to each player $i \in \mathcal{I}$ is then

$$\mathcal{X}_i = \{x_i \in \Omega_i \mid \|x_i - x_j\|_\infty \leq R_{\max},\ j \in \mathcal{N}_i\}\,. \tag{4.11}$$

4.2 Generalized Nash Equilibrium Seeking Algorithms

Before presenting the details of Nash equilibrium seeking algorithms, we invite the reader to notice that (4.6) and (4.9) are indeed convex potential games. Each player $i \in \mathcal{I}$ has been assigned a local utility $U_i(x_i, x_{-i})$ that is continuous in $x \in \Omega$ and convex in the local decision variable x_i, for any fixed x_{-i}. In addition, the local feasible set Ω_i is convex and compact.

The two proposed games can be solved using Algorithm 1, namely, a decentralized version of the preconditioned forward-backward method, presented in [26]. The rationale behind it is that all the players should achieve consensus in terms of constraints satisfaction, once a solution to the game is found. For this purpose, a vector $\lambda \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$ of dual variables, together with auxiliary variables d_i associated to each player, are introduced in order to enforce coupling agreement among agents. At each iteration of Algorithm 1, the next action of each player i is computed as a function of x_{-1} and λ, meaning that convergence to a Nash equilibrium presumes convergence to consensus of the vector λ itself. In the table below, we summarize the main steps of the algorithm.

Algorithm 1 Preconditioned forward-backward (pFB)

Initialization: $x_i^0 \in \Omega_i,\ \lambda^0 \in \mathbb{R}_{\geq 0}$

Iterate until convergence:

Step 1: Local strategy update, $\forall i \in \mathcal{I}$:

$\tilde{x}_i^k \leftarrow x_i^k - \alpha_i\Big(\nabla_{x_i} U_i\left(x_i^k, x_{-i}^k\right) + A_i^\top \lambda^k\Big)$

$x_i^{k+1} \leftarrow \mathrm{proj}_{\Omega_i}\left[\tilde{x}_i^k\right]$

$d_i^{k+1} \leftarrow 2A_i x_i^{k+1} - A_i x_i^k - b_i$

Step 2: Dual variables update :

$\lambda^{k+1} \leftarrow \mathrm{proj}_{\mathbb{R}^{|\mathcal{E}|}_{\geq 0}}\left[\lambda^k + \beta \sum_{i\in\mathcal{I}} d_i^{k+1}\right]$

The operators proj_{Ω_i} and $\mathrm{proj}_{\mathbb{R}^{|\mathcal{E}|}_{\geq 0}}$ denote the projections onto Ω_i and the non-negative $|\mathcal{E}|$-dimensional orthant, respectively.

Chapter 5

Autonomous Online Distributed Formation Control

Feedback Equilibrium Seeking (FES) is an extension of Feedback Optimization (FO) [27], [19] that aims at designing a robust feedback controller to steer a dynamical system to the (a-priori unknown) solution trajectory of a time-varying generalized equation (GE) [12]. GEs can model a wide spectrum of useful objectives in non-cooperative games, including (generalized) Nash Equilibria, as well as constrained optimization problems. In this chapter, which is largely inspired by [12], we tailor the control framework of FES to the problem of optimal relay placement discussed so far. The interested reader is referred to this paper for a complete overview on this subject.

5.1 Feedback Equilibrium Seeking Control

The FES control paradigm addresses general non-linear, continuous-time system

$$\dot{x}(t) = f(x(t), u(t), w(t)) \tag{5.1a}$$
$$y(t) = g(x(t), w(t)) \tag{5.1b}$$

where $x \in \mathcal{L}^{n_x}$ is the state, $u \in \mathcal{L}^{n_u}$ the control input, $w \in \mathcal{L}^{n_w}$ an external disturbance and $y \in \mathcal{L}^{n_y}$ the output of the plant. $\mathcal{L}^n : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is used to denote an n-dimensional signal.

A series of preliminary assumptions asking for local (global) Lipschitz continuity of f (g), compactness and convexity of $\mathcal{U} \subset \mathbb{R}^{n_u}$ and $\mathcal{W} \subset \mathbb{R}^{n_u}$, as well as pre-stabilization of (5.1) is necessary in order to justify some technical aspects that are important when discussing properties of the closed-loop interconnection with the discrete algorithm.
From these assumptions we get to have existence of a steady-state input-output function

$$h(u, w) = g(x_{ss}(u, w), w) \tag{5.2}$$

where $x_{ss} : \mathcal{U} \times \mathcal{W} \to \mathbb{R}^{n_x}$ is a continuously differentiable steady-state map such that $\forall u \in \mathcal{U}$, $\forall w \in \mathcal{W}$, $f(x_{ss}(u, w), u, w) = 0$.

Unlike FO, in FES parameterized GEs are considered, as they represent a better tool to encode efficiency for a variety of steady-state operating conditions:

$$0 \in F(u, y) + \mathcal{B}(u) \tag{5.3a}$$
$$y = h(u, w) \tag{5.3b}$$

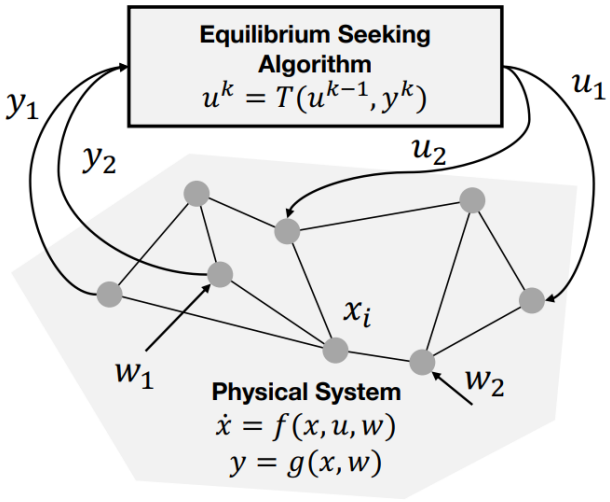

Figure 5.1: In FES, measurements of a dynamical system are fed into an equilibrium seeking algorithm [12]

with $F : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_u}$ being continuously differentiable and $\mathcal{B} : \mathcal{U} \rightrightarrows \mathbb{R}^{n_u}$ a set-valued map. The goal is then to design an output feedback controller that steers the physical system (5.1) to the trajectories $u^*(w)$ and $y^*(w)$ solving eq. (5.3) where:

$$\begin{bmatrix} u^*(w) \\ y^*(w) \end{bmatrix} = \left\{ \begin{matrix} u \in \mathcal{U}, \\ y \in \mathbb{R}^{n_y} \end{matrix} \middle| \begin{matrix} F(u, y) + \mathcal{B}(u) \ni 0, \\ y = h(u, w) \end{matrix} \right\} \tag{5.4}$$

is the solution-to-parameter mapping and u^*, y^* are single-valued mappings.

Assume also that we can measure y, but not w, which is reasonable since exogenous disturbances may be unknown or simply difficult to measure in a realistic settings. Basically, we are dealing with a continuous-time system which we wish to drive to efficient steady-state (or slowly varying) operating points, subject to bounded, unmeasured exogenous disturbances. Exact evaluation of the input-output map h is not possible, hence the need for output measurements y.

At high level, the rationale behind FES is to implement an iterative discrete-time algorithm for solving generalized equations in closed-loop with (non-linear) continuous-time dynamical systems. Specifically, the algorithm is fed with output measurements y from the plant and returns control commands u obtained by solving (5.3) [12].
Consider one algorithm that linearly converges to the solution $u^*(w)$ of (5.3):

$$u^{k+1} = T(u^k, h(u^k, w)), \quad \forall k \in \mathbb{N} \tag{5.5}$$

where $T : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_u}$ is a mapping to the next iterate and w is a constant, measured disturbance. Measurements y are then incorporated in the algorithm to replace h, since w is neither measurable nor constant, using a sampled-data strategy of period $\tau > 0$ combined with a zero-order hold. This results in the hybrid closed-loop interconnection

$$\Sigma_1^s : \begin{cases} \dot{x}(t) = f(x(t), u(t), w(t)) \\ y(t) = g(x(t), w(t)) \end{cases} \tag{5.6a}$$

$$\Sigma_2^s : \begin{cases} u^k = (1 - \varepsilon)u^{k-1} + \varepsilon T(u^{k-1}, y(t^k)) \\ u(t) = u^k, \quad \forall t \in [t^k, t^{k+1}) \end{cases} \tag{5.6b}$$

where $t^k = k\tau$ is the k-th sampling instant and $\varepsilon \in (0, 1]$ is regularization parameter.

In the next section we apply the FES paradigm to distributed robotic swarms coordination and focus on the case of optimal relay placement in wireless networks. The reader is invited from now to note that the use of discrete-time algorithms is particularly important in order to express inter-agent communication, which is naturally a discrete process.

5.2 Coordination of Unicycle Robots through FES

Let us go through the loop in fig. 1.1 and provide a rigorous mathematical formulation of the hybrid system, as in (5.6). To avoid confusion, notation is kept consistent with the one presented in previous chapters.

As before, take a formation of n robotic agents and m mobile users defining a connected undirected graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Fix again a set of q paths $\mathcal{P} := \{\mathcal{P}_1, \dots, \mathcal{P}_q\}$ each connecting one pair of nodes in $\mathcal{J}$ through intermediate relays. Note that one relay can easily be included in more that one path $\mathcal{P}$, and that such paths are fixed by the assumption of time-invariant graph. Recall then the global objective: place the intermediate relays as close as possible to the line connecting two ground stations, at equal distances. This applies to each path in $\mathcal{P}$.

Recall that in section 3.1 we introduced the position commands

$$\overline{\mathbf{r}}_i = \begin{bmatrix} \overline{a}_i \\ \overline{b}_i \end{bmatrix}, \quad \forall i \in \mathcal{I}, \tag{5.7}$$

to express a desired target configuration inside an obstacle-free rectangular region for the i-th unicycle. Likewise, the game formulation in (4.6) is optimized at a set of efficient two-dimensional target relay positions. Since motion is only allowed to happen in a compact, convex region, so will be the action set Ω_i associated to the player in $\mathcal{I}$.

Based on this, we now define the output to the FES-based path planner $u = \overline{\mathbf{r}}$ as a stacked vector of actions for the players $i \in \mathcal{I} := \{1, \dots, n\}$, with $\overline{\mathbf{r}} = [\overline{\mathbf{r}}_1^\top, \dots, \overline{\mathbf{r}}_n^\top]^\top$ and $u \in \mathcal{U} \subset \mathbb{R}^{2n}$. By doing this we identify the output of the upper-level controller and the reference signals that are fed to the pre-stabilized system.

The closed-loop interconnection of i-th dynamical system and its local controller (3.3), when it is given the reference $u_i = \overline{\mathbf{r}}_i$, reads as

$$\begin{aligned} \dot{a}_i &= (k_1 \|\mathbf{r}_i - u_i\|_2 \cos(\phi_i(u_i))) \cos\theta_i + w_{a,i} \\ \dot{b}_i &= (k_1 \|\mathbf{r}_i - u_i\|_2 \cos(\phi_i(u_i))) \sin\theta_i + w_{b,i} \\ \dot{\theta}_i &= -k_1 \cos(\phi_i(u_i)) \sin(\phi_i(u_i)) - k_2 \phi_i(u_i) \end{aligned} \tag{5.8}$$

with $w_i = [w_{x,i} \;\; w_{y,i} \;\; w_{\theta,i}]^\top$ being a vector of additive disturbances accounting for wind gusts that influence the position and the heading of the robot. Too see how ϕ_i depends of the reference position u_i, the reader is referred to section 3.1.

By setting to zero the left-hand side of eq. (5.8) one can get the steady-state input-output $h_i(u_i, w_i)$, relative to player i. In what follows, we avoid computing a closed form of h_i for the system that is subject to external perturbations w_i of this kind. Instead, we notice that in the disturbance-free scenario $h_i(u_i) = u_i$.

We are left with a bunch of self-interested agents, each inducing a steady-state mapping $h_i(u_i, w_i)$ and a utility function $U_i(u_i, h(u, w))$, where $h = [h_i]_{i \in \mathcal{I}}$ is the system wide map. Since each player wants to optimize its own objective we obtain a collection of n-interdependent optimization problems that constitute the convex potential game (see section 4.1.1)

$$\forall i \in \mathcal{I}: \quad \min_{u_i \in \mathcal{U}_i} \quad \{-U_i(u_i, y) \mid y = h(u, w)\}. \tag{5.9}$$

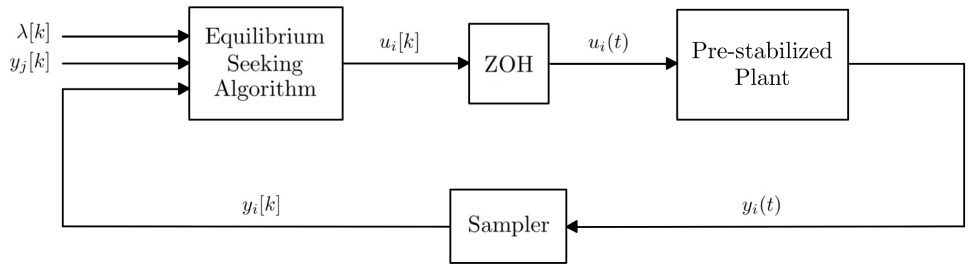

Figure 5.2: Physical system in closed-loop with an equilibrium seeking algorithm.

Sampled measurements of y are then incorporated in algorithm 1 to replace $h(u_u, w)$, as the steady-state may be difficult to reduce to a compact closed form, like in our case, or influenced by unmeasurable disturbances. The same applies for the solution $u^*(w)$ to the GEs in (5.3), thus motivating the use of discrete algorithms for numerical computation. In our setting each player will be measuring the relative xy-position of neighbouring ground stations and robotic relays.
To conclude, we close the loop between the physical system and the equilibrium seeking algorithm, thus obtaining the hybrid system (5.6)

$$\Sigma_1^s : \begin{cases} \dot{x}_i(t) = f(x(t), u(t), w(t)) \\ y_i(t) = x_i(t) \end{cases} \tag{5.10a}$$

$$\Sigma_2^s : \begin{cases} u_i^k = (1-\varepsilon)u_i^{k-1} + \varepsilon \mathrm{proj}_{\Omega_i}\left[u_i^{k-1} - \alpha_i\big(\nabla_{x_i}\big(-U_i(u_i^{k-1}, y)\big) + A_i^\top \lambda^{k-1}\big)\right] \\ d_i^k = 2A_i u_i^k - A_i u_i^{k-1} - b_i \\ \lambda^k = \mathrm{proj}_{\mathbb{R}^{|\mathcal{E}|}_{\geq 0}}\left[\lambda^{k-1} + \beta \sum_{i \in \mathcal{I}} d_i^k\right] \\ u_i(t) = u_i^k, \quad \forall t \in [t^k, t^{k+1}) \end{cases} \tag{5.10b}$$

where $f(x(t), u(t), w(t))$ summarizes the dynamics in eq. (5.8).

5.3 Coordination of Aerial Robots through FES

Take now a next-hop graph connecting $m = 2$ stations exchanging cellular traffic through n intermediate quadcopters. One could argue that such network structure does not account for possible failure of an intermediate node, which is an important issue in practice. However, note that we have no interest whatsoever in defining a good starting graph topology; instead we want to make sure that the multi-agent system can converge to an optimal final configuration while satisfying a number of coupling constraints.

For the optimal placement problem to be well posed a relay must maintain its optimal position once it is reached. Consider the quadrotor i: it can keep a certain position only when it hovers, namely when the roll ϕ_i and pitch θ_i angles are close to zero. Variations in ϕ_i and/or θ_i would indeed be responsible for lateral motion. This means that the main force acting on the quadrotor has to be a total vertical thrust T_i, which is equal in magnitude to the gravity vector $\boldsymbol{G}$.
Therefore we can express a desired reference configuration for i, which would normally be an element of $\mathbb{R}^{12}$, in terms of the desired position of its CoG, i.e., $\overline{\boldsymbol{\xi}}_i \in \mathbb{R}^3$, by setting $\overline{\boldsymbol{\eta}}_i = \overline{\mathbf{v}}_i^B = \overline{\boldsymbol{\nu}}_i = \mathbf{0}_3$, where $\mathbf{0}_3$ is the zero element of $\mathbb{R}^3$. This condition enforces near-hovering conditions throughout the entire mission.

As before, we restrict motion to happen in a convex, closed environment without obstacles. In our setting, actions correspond to position coordinates in this environment. Since every player $i \in \mathcal{I}$ has been assigned a compact, convex action set Ω_i of 3D positions, and a local utility

$U_i : \Omega \to \mathbb{R}$ that is convex on the joint action space Ω, then (4.9) is a convex potential game. Its solution, i.e., the generalized Nash equilibrium, describes the unique optimal formation shape in the 3D space.

Let us now define the output $u \in \mathcal{U}$ of the high-level guidance controller as a tall vector collecting the actions of all players. In particular, the strategy of the i-th player is

$$u_i = \overline{\boldsymbol{\xi}}_i, \quad \forall i \in \mathcal{I}. \tag{5.11}$$

If we call the reference configuration of the i-th quadrotor as $\overline{x}_i(t, u_i) = [\overline{\boldsymbol{\xi}}_i^\top \ \mathbf{0}_9^\top]^\top$, the lower-level MPC solves the following target tracking problem:

$$\begin{aligned}
\min_{U} \quad & \sum_{j=0}^{N-1} \|\mathbf{x}_j - \overline{x}_i(t, u_i)\|_Q + \|\mathbf{u}_j - \overline{\mathbf{u}}\|_R + \|\mathbf{u}_j - \mathbf{u}_{j-1}\|_W \\
\text{s.t.} \quad & \mathbf{x}_{j+1} = f(\mathbf{x}_j, \mathbf{u}_j), && \forall j \in \{0, \dots, N-1\} \\
& \mathbf{x}_j \in \mathcal{X}, && \forall j \in \{0, \dots, N-1\} \\
& \mathbf{u}_j \in \mathcal{U}, && \forall j \in \{0, \dots, N-1\} \\
& \mathbf{x}_N = \overline{x}_i(t, u_i) \\
& \mathbf{x}_0 = \mathbf{x}(k)
\end{aligned} \tag{5.12}$$

where N is length of the prediction horizon, P, Q, W are matrices of weights and $\overline{\mathbf{u}}$ is a steady-state control input necessary to maintain hovering conditions. In eq. (5.12) we dropped the subscript i to make the notation easier.

Once again we take measurements y of $\mathbf{x} = [\mathbf{x}_1^\top \ \dots \ \mathbf{x}_n^\top]^\top$, where $\mathbf{x}$ denotes the state of all players, and incorporate them in the discrete ES algorithm. Finally, by closing the loop we obtain an hybrid system analogous to that in eq. (5.10).

Chapter 6

Simulation Results

In this chapter we validate the FES paradigm for distributed control of robotic swarms using simulation examples. As in the rest of this report, we divide its content in two parts and start by illustrating our work on coordination of a network of unicycle robots.

6.1 Coordination of a Network of Unicycle Relays

We consider a formation of $n = 6$ unicycle relays, each of them implementing the IEEE 802.11ah wireless standard, and $m = 3$ mobile base stations. The use of lower frequencies, that are around 900MHz, is suitable for multi-robot application as it ensures long range inter-agent communication and better penetration through obstacles. In our setting the relays operate in a 20m×20m indoor environment with no obstacles and have perfect knowledge of the distance-dependant path loss parameters $\theta_{\text{PL}} = [\alpha_{\text{SNR,PL}} \;\; n_{\text{PL}}]^\top$ that are responsible for the quality of communication, where $\alpha_{\text{SNR,PL}}$ is a coefficient related to the transmitting power of the antenna, and n_{PL} is the power fall-off rate.
Thus we focus on a deterministic setting and neglect complicated propagation effects like shadowing and multipath fading: extension to such cases presumes taking measurements of the channel quality for a specific environment. Methods to estimate the channel parameters from measurements, as well as communication-aware path planning in realistic environments are found in [4], [8], and [20]. The values we used for θ_{PL} are listed in table (6.1) and give strongly convex utility functions (4.5). In general, one could choose values of $\alpha_{\text{SNR,PL}}$ as a function of the fall-off rate in order to enforce a minimum received SNR among the robots. The value of $\alpha_{\text{SNR,PL}}$ can

θ_{PL}	Value
$\alpha_{\text{SNR,PL}}$	1.25
n_{PL}	2.32

Table 6.1: Path loss parameters.

be chosen arbitrarily as it depends on the transmitting power of the antenna mounted by the relay. We fixed it such that any robot or ground station falling inside the $\|\cdot\|_{\infty,i}$ norm ball of radius $R_{\max}$ centered in the i-th relay's position has a minimum received signal to noise ratio $\Upsilon_{\text{SNR,PL}} \approx 4.5$dB. The values will then satisfy the sufficient conditions regarding convexity of potential function ϕ, as stated in [8].
For the local non-linear controller, we selected control gains as in table (6.2), which provide a good disturbance rejection.
The maximum allowed distance between two relays is set to $R_{\max} = 8$m, and the adopted sampling period is $\tau = 0.1$s. Moreover, all the robotic relays have been given the same value for

	Value
k_1	6
k_2	1.5

Table 6.2: Gains for the non-linear unicycle controller.

the step-sizes in Algorithm 1 (pFB), namely, $\alpha_i = 1.0,\ \forall i \in \mathcal{I},\ \beta = 0.02$.
We considered the initial deployment

	TX	RX_1	RX_2	Player 1	Player 2	Player 3	Player 4	Player 5	Player 6
$a_{i,0}$	-7	7	0	-3	1	4	0	-4	-6
$b_{i,0}$	-7	-7	9	-3	-3	1	1	4	-0
$\theta_{i,0}$	-	-	-	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$	$\pi/2$

Table 6.3: Starting configuration for the network of unicycle relays.

and model an exogenous disturbance acting on the agent i as a wind field $\mathbf{w}_i = [w_{a,i}\ \ w_{b,i}\ \ 0]^\top$. The disturbance is bounded and enters the dynamics as in eq. (2.3). Fig. (6.1) and fig. (6.2) show convergence to a neighbourhood of the optimal network configuration, i.e., the solution to the generalized potential game designed in Section 4.1.1. Asymptotic convergence to the optimal formation shape would be achieved in the case of known disturbances.

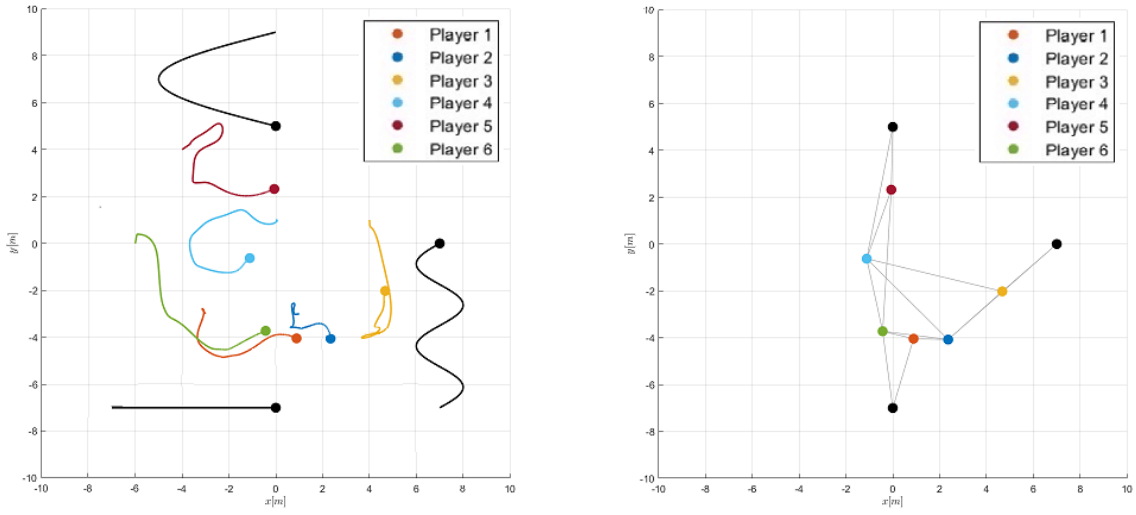

Figure 6.1: Convergence to a neighbourhood of the optimum, i.e., the straight lines connecting the transmitter to the receivers.

6.2 Coordination of a Network of Quadrotors

Consider a formation of $n = 6$ quadrotors moving in a 200m×200m×100m outdoor workspace. The environment is assumed to be obstacle-free, and the minor dynamics affecting the signal propagation (i.e., shadowing and multipath fading) are neglected. Note that these assumptions are not restrictive, since quadrotors can change their altitude and consequently improve the LoS: the presence of buildings will not play an important role and a bit signal travels to the destination

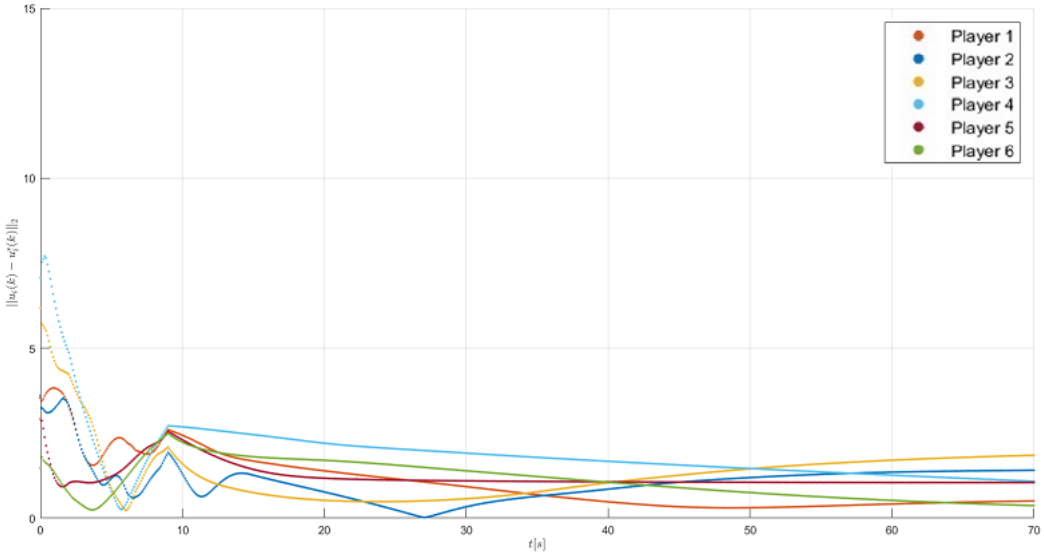

Figure 6.2: Online Nash equilibrium tracking performance of a swarm of unicycle relays. Asymptotic convergence to a neighbourhood of the optimal network configuration.

in almost ideal conditions, thus justifying the simplified game design from Section 4.1.2.

We considered the 3D positions in Table 6.4 as a starting configuration for the multi-agent system. Initial linear velocities, Euler angles, and angular rates are set to zero. This is without loss of generality since we aim at optimizing the position of a quadrotor's CoG in the cartesian space.

	TX	RX	Player 1	Player 2	Player 3	Player 4	Player 5	Player 6
$a_{i,0}$	-70	0	-40	-90	-60	-10	40	10
$b_{i,0}$	-70	0	-70	-50	40	20	15	-40
$c_{i,0}$	30	50	60	50	50	70	30	80

Table 6.4: Starting positions for the network of quadrotors.

Each robotic agent $i \in \mathcal{I}$ implements a local non-linear MPC controller with prediction horizon of length $N = 20$, and sampling period $\tau_{\text{MPC}} = 0.1$s. The weight matrices, which define the quadratic cost function in (3.6), are selected as:

$$Q = \begin{bmatrix} qI_{6\times6} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \end{bmatrix}, \quad R = rI_{4\times4}, \quad W = wI_{4\times4}, \tag{6.1}$$

where $q = 1$ and $r = w = 0.1$: larger differences in the open-loop control sequence with respect to the steady state input $\overline{\mathbf{u}}$, and wilder changes in the control input are allowed if faster convergence to the reference position $\overline{x}_i(t, u_i)$ at the end of the prediction horizon is obtained [24]. To account for actuation limits, we put the following constraints on the squared angular velocity of the propellers:

$$\omega_{k,i}^2 \in [0, 12], \quad \forall k \in \{1, \dots, 4\}, \ \ \forall i \in \mathcal{I}. \tag{6.2}$$

The step-sizes in Algorithm 1 are selected as $\alpha_i = 0.03$ for all $i \in \mathcal{I}$, and $\beta = 0.01$. Auxiliary and dual variables, (i.e., d_i^0 and λ^0) are initialized to zero. As a sample time for the high-level path planner we chose $\tau_{\text{FES}} = 0.1$s.

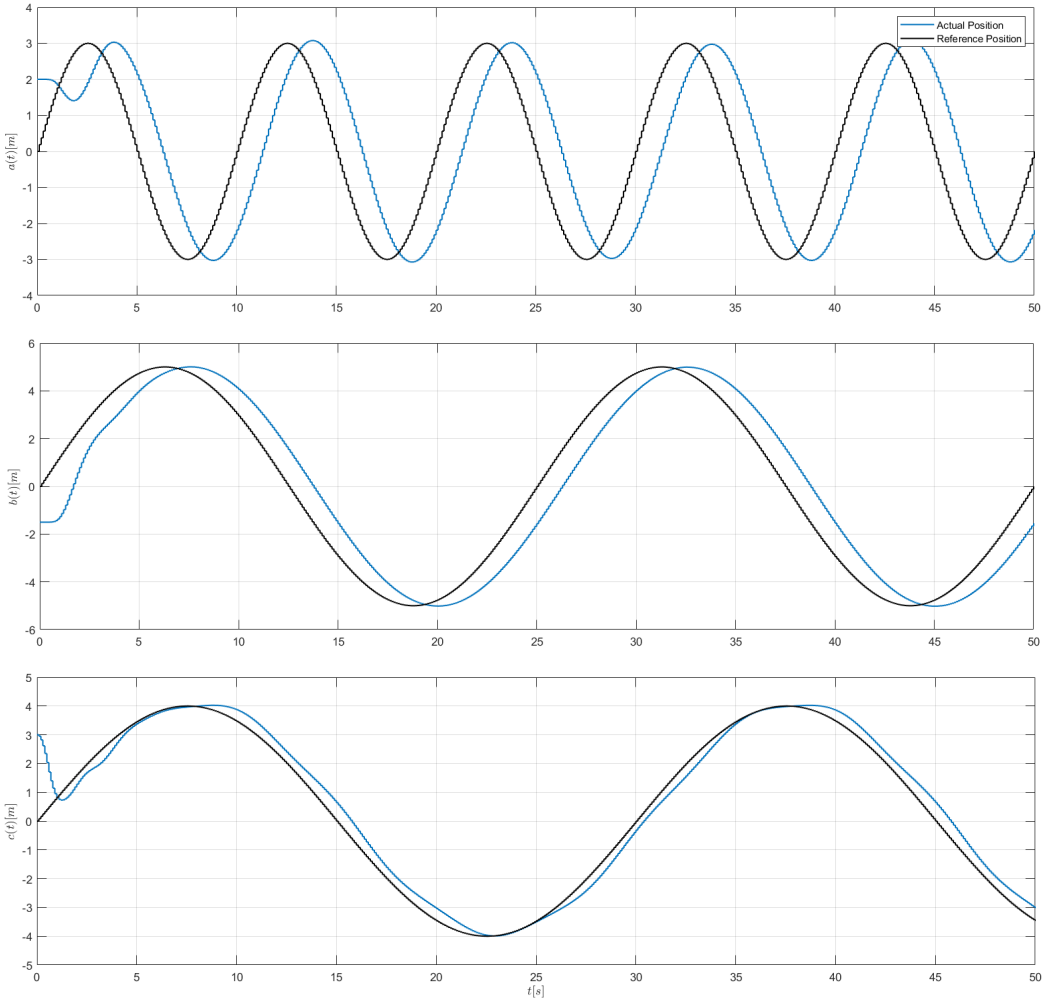

Figure 6.3: Non-linear MPC controller tracking performance.

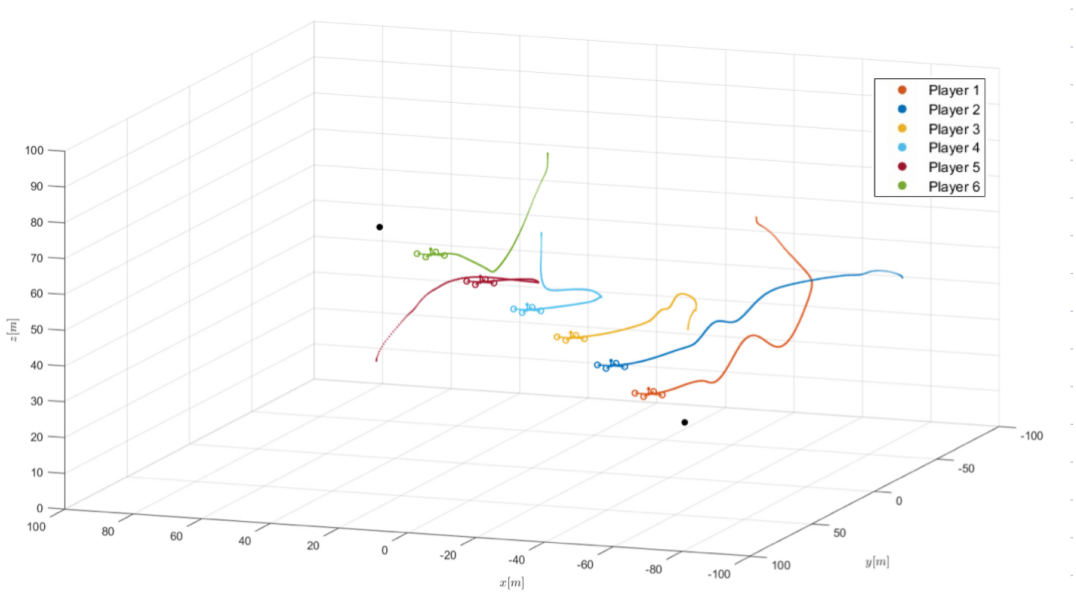

Figure 6.4: The quadrotors place themselves on the line connecting the mobile transmitting and receiving base stations, at equal distances.

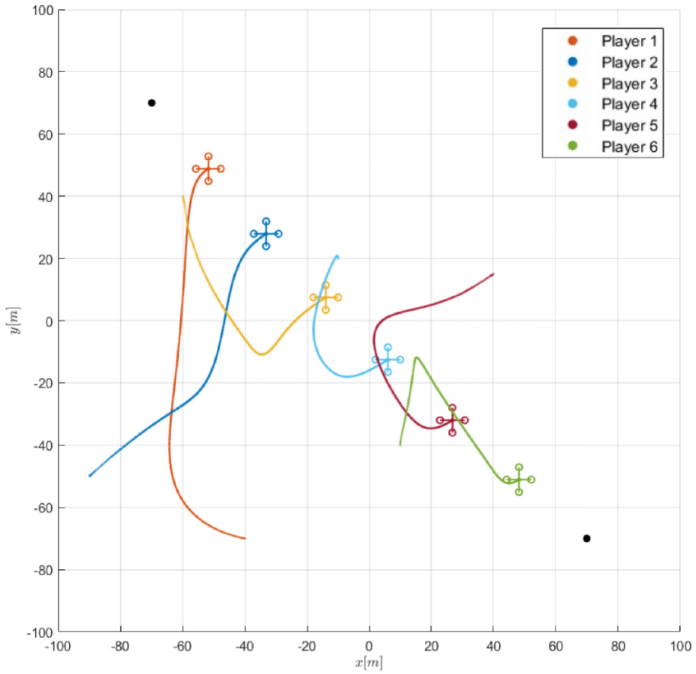

Figure 6.5: Planar view of the optimal configuration for a next-hop network of quadrotors.

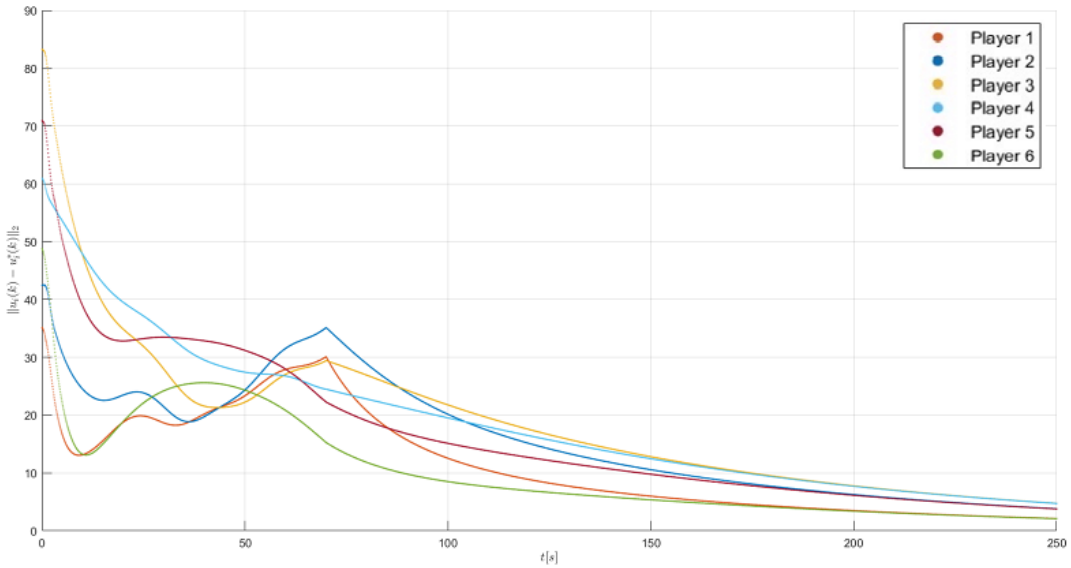

Figure 6.6: Online Nash equilibrium tracking performance for a network of quadrotors. Asymptotic convergence to the optimal configuration.

Chapter 7

Conclusion

This project tailored the theoretical results on FES for two and three-dimensional robotic swarms coordination in quasi-realistic wireless communications scenarios. We have proposed two designs for Online FES-based feedback controllers that guide complex multi-robot systems to an *efficient* steady-state (or slowly varying) operating points and encoded efficiency, i.e., the probability of correct reception and transmission of a digital signal, in terms of parametric GEs, according to the FES control paradigm. Two different local control architectures, namely, a Lyapunov-based controller and a non-linear MPC, both with stability guarantees, have been tested for the controlled continuous-time systems. Finally, asymptotic convergence to the optimal operating points, in the disturbance-free setting, or to a small neighbourhood around them, under the effect of unmeasured perturbations, were assessed through simulation.

Future improvements for our work include optimal parameter tuning in the local controllers and ES algorithm, online estimation of channel quality parameters, generalization to time-varying network structures, derivation of an optimal topology in a learning-in-games fashion.

Bibliography

[1] Guangchi Zhang et al. "Trajectory Optimization and Power Allocation for Multi-Hop UAV Relaying Communications". In: *IEEE Access* 6 (2018), pp. 48566–48576.

[2] Yong Zeng, Rui Zhang, and Teng Joon Lim. "Throughput Maximization for UAV-Enabled Mobile Relaying Systems". In: *IEEE Transactions on Communications* 64.12 (2016), pp. 4983–4996.

[3] 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.

[4] Yasamin Mostofi, Mehrzad Malmirchegini, and Alireza Ghaffarkhah. "Estimation of communication signal strength in robotic networks". In: (2010), pp. 1946–1951.

[5] 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.

[6] Weisi Guo and Tim O'Farrell. "Relay Deployment in Cellular Networks: Planning and Optimization". In: *IEEE Journal on Selected Areas in Communications* 31.8 (2013), pp. 1597–1606.

[7] Weisi Guo et al. "Automated small-cell deployment for heterogeneous cellular networks". In: *IEEE Communications Magazine* 51.5 (2013), pp. 46–53.

[8] Y. Yan. "Co-Optimization of Communication, Motion and Sensing in Mobile Robotic Operations". In: (2016).

[9] Bing Li et al. "Joint Transmit Power and Trajectory Optimization for Two-Way Multi-Hop UAV Relaying Networks". In: *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2020, pp. 1–5.

[10] 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.

[11] Osama M. Bushnaq et al. "Optimal Deployment of Tethered Drones for Maximum Cellular Coverage in User Clusters". In: *IEEE Transactions on Wireless Communications* 20.3 (2021), pp. 2092–2108.

[12] Giuseppe Belgioioso et al. "Sampled-Data Online Feedback Equilibrium Seeking: Stability and Tracking". In: (2021).

[13] S. M. LaValle. *Planning Algorithms*. Available at http://planning.cs.uiuc.edu/. Cambridge, U.K.: Cambridge University Press, 2006.

[14] J. Baillieul et al. *Nonholonomic Mechanics and Control*. Interdisciplinary Applied Mathematics. Springer New York, 2015.

[15] Sung-On Lee et al. "A stable target-tracking control for unicycle mobile robots". In: *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*. Vol. 3. 2000, 1822–1827 vol.3.

[16] Teppo Luukkonen. "Modelling and Control of Quadcopter". In: *Espoo: Aalto University* (2011).

[17] Bruno Siciliano et al. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2010.

[18] Heinz H. Bauschke and Patrick L. Combettes. "Correction to: Convex Analysis and Monotone Operator Theory in Hilbert Spaces". In: *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer International Publishing, 2017.

[19] Adrian Hauswirth et al. *Optimization Algorithms as Robust Feedback Controllers*. 2021.

[20] Usman Ali et al. "An optimal control approach for communication and motion co-optimization in realistic fading environments". In: *2015 American Control Conference (ACC)*. 2015, pp. 2930–2935.

[21] P. Pounds, R. Mahony, and P. Corke. "Modelling and control of a large quadrotor robot". In: *Control Engineering Practice* 18.7 (2010). Special Issue on Aerial Robotics, pp. 691–699.

[22] S. Bouabdallah, A. Noth, and R. Siegwart. "PID vs LQ control techniques applied to an indoor micro quadrotor". In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 3. 2004, 2451–2456 vol.3.

[23] Aws Abdulsalam Najm and Ibraheem Kasim Ibraheem. "Nonlinear PID controller design for a 6-DOF UAV quadrotor system". In: *Engineering Science and Technology, an International Journal* 22.4 (2019), pp. 1087–1097.

[24] Emmanouil Tzorakoleftherakis and Todd D. Murphey. "Iterative Sequential Action Control for Stable, Model-Based Control of Nonlinear Systems". In: *IEEE Transactions on Automatic Control* 64.8 (2019), pp. 3170–3183.

[25] Jason R. Marden, GÃrdal Arslan, and Jeff S. Shamma. "Cooperative Control and Potential Games". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.6 (2009), pp. 1393–1407.

[26] Giuseppe Belgioioso and Sergio Grammatico. "Semi-decentralized generalized Nash equilibrium seeking in monotone aggregative games". In: *IEEE Transactions on Automatic Control* (2021), pp. 1–1.

[27] Andrea Simonetto et al. *Time-Varying Convex Optimization: Time-Structured Algorithms and Applications*. 2020.