

# Semester Project

## Distributed Satellite Formation Control via Feedback Equilibrium Seeking

Leonardo Barcotto  
August 2021

### Advisors

Dr. Giuseppe Belgioioso, Dr. Dominic Liao-McPherson,  
Prof. Florian Dörfler

# Abstract

Distributed Satellite Formation Control via Feedback Equilibrium Seeking

Leonardo Barcotto

MSc Robotics, Systems and Control  
Eidgenössische Technische Hochschule Zürich  
2021

Spacecraft formation flying refers to the problem of coordinating multiple closely located satellites, in order to allow them to behave as a cohesive system. In this report, we employ a game-theoretic approach for controlling satellites clusters in a distributed manner. The relative dynamics between satellites are expressed using the Hill–Clohessy–Wiltshire equations of relative motion. A linear-quadratic regulator is implemented locally on each spacecraft as a low-thrust controller in order to stabilize the motion of the satellites, subject to the atmospheric drag, the moon’s gravity and the  $J_2$  perturbation, and to track the reference orbits commanded by an upper level formation controller. Two coverage coordination games are designed, to model the acquisition and formation-keeping problems in the orbital plane. The (static) games are solved using a distributed generalized Nash equilibrium (GNE) seeking algorithm, and feedback equilibrium seeking (FES) control is implemented for the planar problem.

## **Acknowledgements**

I would like to express my gratitude to my primary supervisors, Dr. Giuseppe Belgioioso and Dr. Dominic Liao-McPherson, for the golden opportunity they gave me to learn so many new things, and for the competence and patience they showed while guiding me throughout this project.



# Contents

<b>List of Figures</b>	<b>7</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Literature Review . . . . .	1
1.2.1 Formation Control . . . . .	1
1.2.2 Robotic Swarms Coordination . . . . .	2
1.2.3 Feedback Equilibrium Seeking (FES) Control . . . . .	2
1.3 Formation Flying Control Architecture . . . . .	3
<b>2 Orbital Dynamics</b>	<b>5</b>
2.1 Reference Frames and Transformations . . . . .	5
2.1.1 Earth Centered Inertial Frame . . . . .	5
2.1.2 Perifocal Frame . . . . .	5
2.1.3 Local Vertical Local Horizontal Frame . . . . .	6
2.2 The Two-Body Problem . . . . .	7
2.2.1 Constants of Orbital Motion . . . . .	8
2.2.2 Shape of a Keplerian Orbit . . . . .	8
2.3 Orbital Perturbations . . . . .	9
2.3.1 Atmospheric Drag . . . . .	9
2.3.2 Third Body Disturbance . . . . .	9
2.3.3 Non-Spherical Gravity ( $J_2$ perturbations) . . . . .	10
2.4 The Hill–Clohessy–Wiltshire Equations of Relative Motion . . . . .	11
2.4.1 The Projected Circular Orbit . . . . .	12
<b>3 Controller Design</b>	<b>15</b>
3.1 The Linear Quadratic Regulator . . . . .	15
3.2 Choice of LQR Weights and Stability Analysis . . . . .	15
<b>4 Coverage Coordination Game</b>	<b>17</b>
4.1 Game Design . . . . .	17
4.1.1 Acquisition Game . . . . .	18
4.1.2 Formation-keeping Game . . . . .	19
4.2 Game Solution . . . . .	19
<b>5 Feedback Equilibrium Seeking Control</b>	<b>21</b>
5.1 General Design Framework . . . . .	21
5.1.1 Problem Setting . . . . .	21
5.1.2 Control Strategy . . . . .	22
5.2 Satellite Formation Control using FES . . . . .	23

<b>6 Simulation Results</b>	<b>27</b>
<b>Bibliography</b>	<b>34</b>

# List of Figures

1.1	Block diagram of the control architecture. . . . .	3
2.1	ECI frame [7]. . . . .	5
2.2	Perifocal coordinate system [7]. . . . .	6
2.3	LVLH frame [7]. . . . .	6
2.4	Orbital angular momentum and eccentricity vector [7]. . . . .	8
2.5	3D ellipse generated by a section of the 3D cylinder when $\alpha_y = \alpha_z$ [11]. . . . .	12
2.6	Geometry of a relative orbit using the HCW equations for relative motion, with circular xy-projection and linear zy-projection [11]. . . . .	13
5.1	Measurements from a physical system are fed into an equilibrium seeking algorithm [2]. . . . .	22
6.1	PCO in LVLH frame for the open-loop acquisition phase. The satellites, starting from the same initial position, equally-space themselves in the orbit. . . . .	28
6.2	Motion along the $x$ , $y$ and $z$ axes of the LVLH frame during the open-loop acquisition phase. . . . .	29
6.3	Control inputs from the lower-level linear quadratic regulators during the open-loop acquisition of the formation, in LVLH frame. . . . .	30
6.4	Close-up of Fig. 6.3 showing the stabilizing control inputs. . . . .	30
6.5	PCO representation in LVLH frame for the open-loop formation-keeping phase. The initial orbit is maintained by the satellites. . . . .	31
6.6	Control inputs from the lower-level linear quadratic regulators during the open-loop formation-keeping phase, in LVLH frame. . . . .	31
6.7	Trajectories of $\ x_k - x^*\ _2$ . The players find the unique generalized Nash Equilibrium. . . . .	32
6.8	Trajectories of the decisions of the players converge to the generalized Nash Equilibrium. . . . .	32
6.9	Control inputs from the lower-level linear quadratic regulators for the closed-loop formation-keeping phase, in LVLH frame. . . . .	33
6.10	Effect of the atmospheric drag on the uncontrolled closed-loop formation-keeping phase. . . . .	33
6.11	PCO in LVLH frame for the controlled closed-loop formation-keeping phase. . . . .	34



# Chapter 1

## Introduction

### 1.1 Introduction

Spacecraft formation flying is the problem of coordinating multiple satellites which allows them to behave as a cohesive system, in order to accomplish goals that would be impossible for a single satellite. Formation flying is characterized by the fact that spacecrafts in a formation operate in close proximity, maintaining a relative geometry, unlike what happens in a constellation, which is a group of satellites working together in orbital planes that cover the entire Earth, providing some planet-wide service.

The problems of acquisition (i.e. establishing the formation) and formation-keeping of such large-scale systems pose a significant challenge in terms of designing suitable and robust control strategies.

In the classical setting, a large number of ground-based stations is available to measure and control the spacecrafts. Thus, the acquisition and formation-keeping problems can be solved in a centralized fashion [1]. However, this centralization is quite unrealistic or undesirable in absence of ground stations (e.g., for planet exploration), or for large-scale formations, since the communication and computational burden of the control law may become unbearable.

Motivated by some recent advances in robotic networks coordination [12], distributed equilibrium seeking algorithms [3] and control [2], in this project we aim at designing a distributed, scalable, robust control strategy for acquisition and formation-keeping in large-scale satellite formations.

### 1.2 Literature Review

In this section we review the state-of-the-art on formation control, robotic swarms coordination and feedback equilibrium seeking control.

#### 1.2.1 Formation Control

A satellite formation is a group of satellites, of similar type and function, designed to work together as a system. Formations are typically made up of numerous small spacecrafts placed in close vicinity to each other and, needless to say, a formation enables objectives that may be difficult or impossible for a single satellite. Formations can be used for various purposes, such as satellite inspection, space assembly and Earth-imaging.

Flying multiple satellites together in formation is a challenging task, especially from the point of view of their control. In this regard, two main phases can be outlined. First, once a cluster of satellites have been deployed in space by a delivery vehicle, it needs to establish the desired formation: this process is referred to as *acquisition*. Then, the relative orbital positions between

satellites must be maintained, even in the presence of orbital disturbances: this is what we call *formation-keeping*. For example, we may want to create a circular formation of equally-spaced satellites moving around Earth in a low-Earth orbit (LEO) and then maintain it despite the effect of atmospheric drag. A third category can also be mentioned, which is reconfiguration of the formation to a different desired geometry.

In order to control groups of satellites, it is typical to connect them to ground stations, which communicate with and control the spacecrafts individually. This results in a centralized control approach. In [1], centralized control using differential drag maneuvers is proposed for the acquisition of a small satellite constellation in low-Earth orbit and a linear program is designed to equally space the satellite in the selected orbital plane.

Note that the methods used for the acquisition and formation-keeping of a constellation can also be recasted for the control of formations.

In this thesis, we address the problem of developing a distributed approach for the acquisition and formation-keeping of a formation. The reasons for such a choice are motivated by the use of large formations with many spacecraft, in which the communication with ground-based stations requires a high bandwidth demand. Moreover, a centralized approach is impossible in situations where ground stations are not available, as happens for example with the exploration of the Moon or Mars. In [20], a linear-quadratic controller for relative formation-keeping of a constellation is developed. A method for autonomous ring formation and station-keeping for a planar constellation of satellites based on the concept of potential function is presented in [16]. In [24] the control problem is addressed by designing a passivity-based distributed control law to arrange a set of satellites into an equally-spaced, planar constellation in a desired circular aerostationary orbit about a planet.

### 1.2.2 Robotic Swarms Coordination

The problem of control and coordination of multirobot systems can be addressed using various approaches. Most of the time, the behavior of the individual agents is driven by their own interests and objectives and, in such a setting, game-based control is a particularly suited tool for the system's regulation [21]. In [14], the authors show how problems of multiple robot coordination can be represented and analysed using cooperative game theory. In [12], the multirobot coordination problem is addressed using two optimization-based approaches, the first one based on distributed optimization and the second based on game-theoretic self-organization, where each robot is endowed with online updating rules designed to enable the swarm to achieve a collective objective. In [9] a distributed coverage control problem on graphs in a game-theoretic setting is studied employing a communication-free setup, in which each agent can sense the presence of other agents and the local graph structure.

### 1.2.3 Feedback Equilibrium Seeking (FES) Control

In this project we aim at designing a distributed, scalable, robust control strategy for the acquisition and formation-keeping of large-scale satellite formations. We basically have a complex dynamical system (a cluster of satellites) and we want to drive it to a desired state (an equally-spaced planar formation). Moreover, we want to keep the system in that state.

There exist typically two different approaches to this problem [2]: optimality, where one wants to operate a system at an optimal steady-state, and game-theoretic solution concepts, such as generalized Nash Equilibria, which are more suitable when dealing with multi-agent systems.

Optimality concepts are further divided into *feedforward optimization*, in which an optimization problem is periodically solved out-of-the-loop and then its solution is fed to a tracking-type controller, and *feedback optimization*, where the control design directly implements optimization

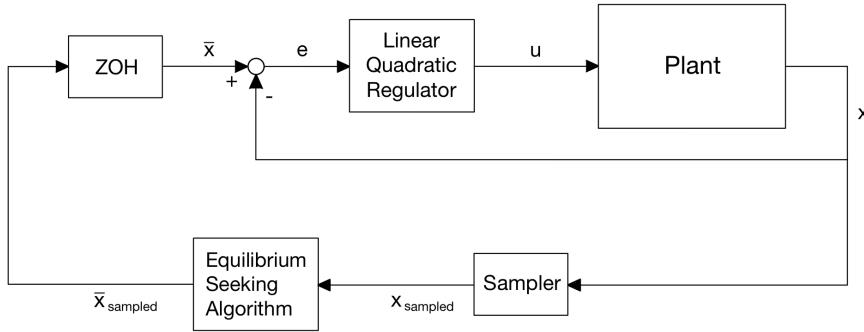


Figure 1.1: Block diagram of the control architecture.

algorithms in closed loop with the physical systems [8]. In [5] the authors present a time-varying convex optimization-based method modified to accomodate measurement feedback for static networks. In [4], a control scheme for feedback optimization of static plants with a discrete-time controller and global convergence guarantees for non-convex problems is introduced. Dynamics of the underlying system is taken into account in [6], where feedback optimization-based methods to steer LTI dynamical systems to the solution of a time-varying convex optimization problem are developed.

Let us now consider the use of game-theoretic solution concepts for the problem of distributed, multi-agent coordination and control. In this setting, a noncooperative game is formulated with agents selfishly trying to optimize their individual cost functions by using locally available information. In this context, one control approach is built on passivity properties. In [10], a distributed control law for NE seeking over networks based on passivity is studied, and convergence to a NE is guaranteed in a disturbance-free setting with single-integrator dynamics agents. The case with external deterministic disturbances and agents with multi-integrator dynamics is studied in [18]. [13] exploits a different control approach, built on Extremum Seeking (ES) algorithms, by designing a discrete-time ES-based controller for distributedly seek NE in games with single or double integrator agents.

### 1.3 Formation Flying Control Architecture

In this section, the general control architecture for the satellite formation control is presented. The block diagram in Fig. 1.1 highlights two feedback loops. An inner loop is closed around the plant, representing the cluster of satellites, and a linear quadratic regulator (LQR) block, whose function is to locally stabilize the motion of the individual satellites, in order to reject the effect of orbital perturbations (see section 2.3) and to track the reference coming from the outer loop. The design of the lower level LQR stabilizers will be discussed in chapter 3.

Then, an outer feedback loop links the stabilized physical system to an upper level formation controller, which coordinates the relative motion of the spacecrafts by embedding them in a coverage coordination game. The game design and the equilibrium-seeking algorithm are treated in chapter 4, while we will talk about feedback equilibrium seeking control in chapter 5.



# Chapter 2

## Orbital Dynamics

In this chapter, we present the equations necessary to model the motion of a spacecraft orbiting a planet and the relative motion of two spacecrafts.

### 2.1 Reference Frames and Transformations

#### 2.1.1 Earth Centered Inertial Frame

The inertial coordinate system used to define the satellites' position and velocity vectors,  $\vec{r}$  and  $\vec{v}$  respectively. As showed in Fig. 2.1, it is centered at the Earth, with the equator as the fundamental plane ( $xy$ -plane). The ECI frame is determined by the basis vectors  $\mathbf{e}_{x,I}, \mathbf{e}_{y,I}, \mathbf{e}_{z,I}$ . The  $\mathbf{e}_{x,I}$  direction is along the vernal equinox, the  $\mathbf{e}_{z,I}$  direction points towards the north pole, and  $\mathbf{e}_{y,I}$  is chosen to complete the right-handed coordinate system, i.e.,  $\mathbf{e}_{y,I} = \mathbf{e}_{z,I} \times \mathbf{e}_{x,I}$ . We will refer to the ECI system using the subscript  $I$ .

#### 2.1.2 Perifocal Frame

The perifocal reference frame, which we will refer to using the subscript  $P$ , is pictured in Fig. 2.2. It is a Cartesian coordinate system centered at the principal body with the orbital plane as the fundamental plane, with unit vectors

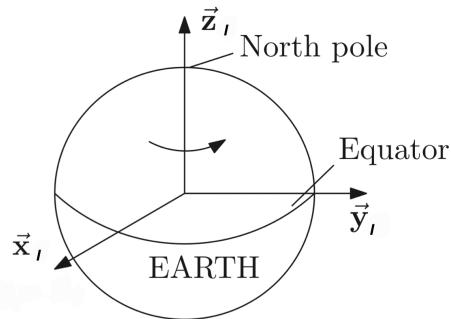


Figure 2.1: ECI frame [7].

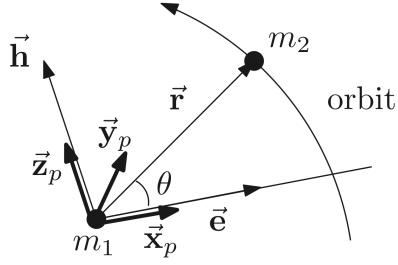


Figure 2.2: Perifocal coordinate system [7].

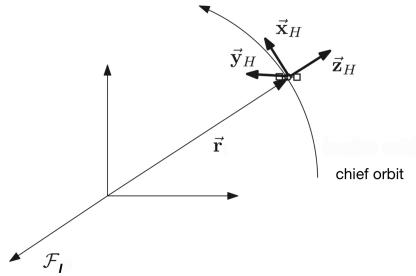


Figure 2.3: LVLH frame [7].

$$\mathbf{e}_{x,P} = \hat{\mathbf{e}}, \quad (2.1)$$

$$\mathbf{e}_{z,P} = \hat{\mathbf{h}}, \quad (2.2)$$

$$\mathbf{e}_{y,P} = \mathbf{e}_{z,P} \times \mathbf{e}_{x,P}, \quad (2.3)$$

where  $\hat{\mathbf{e}}$  and  $\hat{\mathbf{h}}$  are the normalized eccentricity vector and angular momentum defining the orbital plane. For a circular orbit  $\hat{\mathbf{e}} = \vec{0}$  and  $\mathbf{e}_{x,P}$  can be arbitrarily chosen as any inertially fixed vector in the orbital plane [7].

### 2.1.3 Local Vertical Local Horizontal Frame

The Local Vertical Local Horizontal (LVLH) or Hill frame is a rotating frame used to express the position of a *deputy* spacecraft with respect to a *chief* spacecraft. It is depicted in Fig. 2.3 and we will refer to this frame with the subscript  $H$ . It is centered at the rotating chief spacecraft and its basis vectors are defined as follows: the  $\mathbf{e}_{z,H}$  axis is directed as  $\vec{r}$ , the position vector of the chief vehicle relative to the center of the Earth, the  $\mathbf{e}_{y,H}$  axis is along the spacecraft's angular momentum  $\hat{\mathbf{h}}$ , and it is orthogonal to the orbital plane and the  $\mathbf{e}_{x,H}$  axis completes the right handed reference frame and is along a tangent to the orbit, i.e.,

$$\mathbf{e}_{z,H} = \hat{\mathbf{r}}, \quad (2.4)$$

$$\mathbf{e}_{y,H} = \hat{\mathbf{h}}, \quad (2.5)$$

$$\mathbf{e}_{x,H} = -\mathbf{e}_{z,H} \times \mathbf{e}_{y,H}, \quad (2.6)$$

where  $\hat{\mathbf{r}}$  and  $\hat{\mathbf{h}}$  are the normalized position vector and angular momentum of the chief, respectively.

Transformations from the ECI frame to the LVLH frame for position and velocity are given by:

$$\vec{r}_H = C\vec{r}_I, \quad (2.7)$$

$$\dot{\vec{r}}_H = \dot{C}\vec{r}_I + C\dot{\vec{r}}_I, \quad (2.8)$$

where  $C$  is the rotation matrix expressing the rotation from the ECI to the LVLH frame, and  $\dot{C}$  is its time derivative. From [22],

$$C = \begin{bmatrix} \mathbf{e}_{x,H} \cdot \mathbf{e}_{x,I} & \mathbf{e}_{x,H} \cdot \mathbf{e}_{y,I} & \mathbf{e}_{x,H} \cdot \mathbf{e}_{z,I} \\ \mathbf{e}_{y,H} \cdot \mathbf{e}_{x,I} & \mathbf{e}_{y,H} \cdot \mathbf{e}_{y,I} & \mathbf{e}_{y,H} \cdot \mathbf{e}_{z,I} \\ \mathbf{e}_{z,H} \cdot \mathbf{e}_{x,I} & \mathbf{e}_{z,H} \cdot \mathbf{e}_{y,I} & \mathbf{e}_{z,H} \cdot \mathbf{e}_{z,I} \end{bmatrix}. \quad (2.9)$$

Computing the time derivatives of the basis vectors, we can easily obtain the transformation matrix derivative. Again, from [22],

$$\frac{d}{dt}(\mathbf{e}_{z,H}) = \frac{1}{r}[\vec{v} - (\hat{\mathbf{r}} \cdot \vec{v})\hat{\mathbf{r}}], \quad (2.10)$$

$$\frac{d}{dt}(\mathbf{e}_{y,H}) = \frac{1}{h}[\dot{\hat{\mathbf{h}}} - (\hat{\mathbf{h}} \cdot \dot{\hat{\mathbf{h}}})\hat{\mathbf{h}}] = \mathbf{0}, \quad (2.11)$$

$$\frac{d}{dt}(\mathbf{e}_{x,H}) = \hat{\mathbf{h}} \times \frac{d}{dt}(\mathbf{e}_{z,H}). \quad (2.12)$$

Note that Eq. (2.11) reduces to  $\mathbf{0}$  because  $\dot{\hat{\mathbf{h}}} = \vec{r} \times \vec{a}$ , where  $\vec{a} = \ddot{\vec{r}}$ , and for two-body motion we know that  $\vec{r} \parallel \vec{a}$ , which is,  $\vec{r}$  is parallel to  $\vec{a}$ .

## 2.2 The Two-Body Problem

The Keplerian two-body problem regards the motion of one body about another under the influence of their mutual gravitation. Even though it considers the case where no other forces are involved in the motion of the two bodies, it constitutes a good model for spacecraft orbiting planets.

To start, consider two point masses,  $m_1$  and  $m_2$ , which represent the mass of the primary body (the planet) and the spacecraft's mass, respectively. We assume that  $m_1 \gg m_2$ . Let us define an inertial reference frame,  $\mathcal{F}_I$ , and let  $\vec{r}_1$  and  $\vec{r}_2$  be the positions of the two point masses from the origin of  $\mathcal{F}_I$ . By expressing the position of mass  $m_2$  with respect to mass  $m_1$  as  $\vec{r}_{21} = \vec{r}_2 - \vec{r}_1$  and applying Newton's second law to find the equations of motion of the two masses, it is easy to find the differential equation governing the motion of the spacecraft  $m_2$  with respect to the planet  $m_1$  as

$$\ddot{\vec{r}} = -\frac{Gm_1}{|\vec{r}|^3}\vec{r}, \quad (2.13)$$

where the subscript "21" has been dropped ( $\vec{r} = \vec{r}_{21}$ ) and  $G$  is the universal gravitational constant. Defining the gravitational constant of the principal body as  $\mu = Gm_1$  we can now state the Keplerian two-body orbital equation of motion

$$\ddot{\vec{r}} = -\frac{\mu}{|\vec{r}|^3}\vec{r}, \quad (2.14)$$

which gives the motion of the mass  $m_2$  about the primary body  $m_1$ .

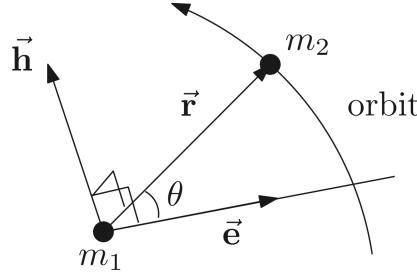


Figure 2.4: Orbital angular momentum and eccentricity vector [7].

### 2.2.1 Constants of Orbital Motion

Let us now look at two constants of motion, namely the *orbital angular momentum* and the *eccentricity vector*, which will help us derive properties of the Keplerian orbital motion we care about for this project.

The *orbital angular momentum* is defined as

$$\vec{h} \triangleq \vec{r} \times \vec{v}, \quad (2.15)$$

where  $\vec{v} = \dot{\vec{r}}$  is the orbital velocity. Taking the inertial time-derivative of  $\vec{h}$ , we find out that the orbital angular momentum is constant with respect to the inertial frame  $\mathcal{F}_I$ :

$$\begin{aligned} \dot{\vec{h}} &= \dot{\vec{r}} \times \vec{v} + \vec{r} \times \dot{\vec{v}} \\ &= \vec{v} \times \vec{v} + \vec{r} \times \ddot{\vec{r}} \\ &= \vec{r} \times \ddot{\vec{r}} \\ &= -\frac{\mu}{|\vec{r}|^3} (\vec{r} \times \vec{r}) = \vec{0}, \end{aligned}$$

where Eq. (2.14) was used. Moreover, by definition of cross-product of vectors, Eq. (2.15) tells us that the position vector  $\vec{r}$  evolves on a plane in inertial space remaining always perpendicular to  $\vec{h}$ . We can conclude that the Keplerian orbital motion is planar, with the orbital plane specified by the angular momentum  $\vec{h}$ .

The second constant of motion we examine is the *eccentricity vector*

$$\vec{e} = \frac{\vec{v} \times \vec{h}}{\mu} - \frac{\vec{r}}{|\vec{r}|}, \quad (2.16)$$

with zero inertial time derivative. The eccentricity vector  $\vec{e}$  is not only constant but it also lies in the orbital plane, since it is defined as the difference between vectors which are perpendicular to  $\vec{h}$ . So,  $\vec{e}$  is fixed in the orbital plane. We define the *eccentricity* of an orbit as  $e = |\vec{e}|$ .

### 2.2.2 Shape of a Keplerian Orbit

To conclude this section, let us briefly talk about the shape that a Keplerian orbit can take, which is equivalent to describing how  $m_2$  moves with respect to  $m_1$ .

First, let us call the distance between the two masses  $r = |\vec{r}|$  and introduce  $\theta$ , called the *true anomaly*, the angle that the position vector  $\vec{r}$  makes with  $\vec{e}$ . In [11], the equation determining the Keplerian orbits is derived as:

$$r = \frac{p}{1 + e \cos \theta}, \quad (2.17)$$

where  $p = h^2/\mu$  is called the *semilatus rectum*. Eq. (2.17) is all it's needed to uniquely determine the shape of a Keplerian orbit. In particular, here we especially care about two types of orbits: circular and elliptical.

A circular orbit is trivially obtained when  $e = 0$ . In this case, from (2.17) it is immediately evident that  $r = p = \text{constant}$ , which means that the distance between the two masses remains always the same, and  $m_2$  follows a circular orbit of radius  $r$  around  $m_1$ .

The elliptical shape of an orbit instead is obtained for  $0 < e < 1$ . To see this clearly, Eq. (2.17) is rearranged and written in perifocal frame (see section 2.1.2) as in [7]. We obtain

$$\frac{(x_P + ae)^2}{a^2} + \frac{y_P^2}{(1 - e^2)a^2} = 1, \quad (2.18)$$

where  $a = \frac{p}{1-e^2}$ ,  $x_P = r \cos \theta$  and  $y_P = r \sin \theta$ , which is the equation of an ellipse with semi-major axis  $a$ , semi-minor axis  $a\sqrt{(1 - e^2)}$  and center  $(-ae, 0)$ .

## 2.3 Orbital Perturbations

As one can imagine, the Keplerian two-body motion is an idealized motion. In reality, there are several factors that contribute to perturb the motion of the spacecraft and to make it deviate from the Keplerian orbit. In this section we will present the equations necessary to model some, but not all, of these factors, in the case of a geocentric orbit.

### 2.3.1 Atmospheric Drag

To start, let us consider the atmospheric drag force. This force is experienced by a satellite moving along its orbit, and it acts against its velocity relative to the atmosphere [1]. The mass-specific acceleration due to the atmospheric drag force is given by:

$$\vec{a}_{atmdrag} = -\frac{1}{2} \frac{C_D A}{m} \rho |\vec{v}_{rel}| \cdot \vec{v}_{rel}, \quad (2.19)$$

where  $C_D$  is the satellite drag coefficient,  $A$  is the surface area exposed to incident stream,  $m$  is the satellite mass,  $\rho = \rho(|\vec{r}|)$  is the atmospheric density at the satellite position and  $\vec{v}_{rel}$  is the velocity of the satellite relative to the atmosphere.

The satellite velocity vector in relation to the atmosphere is approximated in [1] based on the assumption that the atmosphere rotates with the same velocity as that of the Earth's rotation:

$$\vec{v}_{rel} = \vec{v}_{sat} - \omega_E \times \vec{r}_{sat}, \quad (2.20)$$

where  $\vec{v}_{sat}$  is the satellite velocity vector,  $\vec{r}_{sat}$  is the satellite position vector, and  $\omega_E$  is the Earth's angular velocity about its axis.

The atmospheric drag force causes gradual orbit decay and it is the main non-gravitational effect on satellites in Low-Earth orbits.

### 2.3.2 Third Body Disturbance

The motion of a spacecraft orbiting a planet is also influenced by the presence of the other celestial bodies. For instance, a spacecraft moving around Earth will have its motion affected by the moon and the sun, and also by the other planets, in a smaller measure.

In this project, perturbing acceleration due to the presence of the moon is considered. In [19], such an acceleration is given as

$$\vec{a}_{moon} = G m_{moon} \left( \frac{\vec{r}_{moon} - \vec{r}}{|\vec{r}_{moon} - \vec{r}|^3} - \frac{\vec{r}_{moon}}{|\vec{r}_{moon}|^3} \right), \quad (2.21)$$

where  $m_{moon}$  is the mass of the moon and with  $\vec{r}_{moon}$  we refer to the position vector of the moon with respect to the Earth.

### 2.3.3 Non-Spherical Gravity ( $J_2$ perturbations)

Now, the effects of perturbative accelerations due to a non-spherical primary body are presented. These effects are very important for Earth-orbiting satellites.

In general, a force  $\vec{f}$  per unit mass can be obtained from a potential function  $\phi$  such that  $\vec{f} = \vec{\nabla}\phi$ , where  $\vec{\nabla}$  is the gradient operator. If we consider a point  $\vec{r}$  outside the body, with associated spherical coordinates

$$x = r \cos \delta \cos \lambda, \quad (2.22)$$

$$y = r \cos \delta \sin \lambda, \quad (2.23)$$

$$z = r \sin \delta, \quad (2.24)$$

the perturbing gravitational potential of the primary body at  $\vec{r}$  is given in [7] by the spherical harmonic formula as:

$$\begin{aligned} \Phi_p(\vec{r}) = & \frac{Gm_1}{r} \left[ - \sum_{n=2}^{\infty} J_n \left( \frac{R_e}{r} \right)^n P_n(\sin \delta) \right. \\ & \left. + \sum_{n=2}^{\infty} \sum_{m=1}^n \left( \frac{R_e}{r} \right)^n P_{n,m}(\sin \delta) [C_{n,m} \cos(m\lambda) + S_{n,m} \sin(m\lambda)] \right], \end{aligned} \quad (2.25)$$

where  $r = |\vec{r}|$  is the magnitude of the position vector,  $R_e$  is some normalizing radius of the main body,  $J_n$ ,  $C_{n,m}$  and  $S_{n,m}$  are experimentally determined coefficients, obtained from satellite observations,  $P_n(x)$  are the Legendre polynomials and  $P_{n,m}(x)$  are the associated Legendre functions. The derivation of this formula is quite involved, for the details we refer the reader to [7].

When the main body is rotationally symmetric about an axis, as in the case of the Earth, we have  $C_{n,m} = S_{n,m} = 0$ , and Eq. (2.25) simplifies to

$$\Phi_p(\vec{r}) = - \frac{Gm_1}{r} \sum_{n=2}^{\infty} J_n \left( \frac{R_e}{r} \right)^n P_n(\sin \delta). \quad (2.26)$$

For the Earth, the dominant perturbing effect is the  $J_2$  term, which is a result of Earth's oblate shape. We have

$$J_2 = 1.083 \times 10^{-3} \quad (2.27)$$

and the perturbing potential given by  $J_2$  effects only is

$$\Phi_p(\vec{r}) = - \frac{\mu}{r^3} J_2 R_e^2 \left( \frac{3}{2} \sin^2 \delta - \frac{1}{2} \right). \quad (2.28)$$

Given the potential, the mass-specific acceleration due to this perturbative force can be obtained by taking its gradient. If we consider ECI coordinates, we have, again from [7],

$$\vec{a}_{J_2} = \frac{3\mu J_2 R_e^2}{2r^5} \left[ \left( 5 \frac{(\vec{r} \cdot \mathbf{e}_{z,I})^2}{r^2} - 1 \right) \vec{r} - 2 (\vec{r} \cdot \mathbf{e}_{z,I}) \mathbf{e}_{z,I} \right]. \quad (2.29)$$

## 2.4 The Hill–Clohessy–Wiltshire Equations of Relative Motion

The Hill-Clohessy-Wiltshire (HCW) equations are a set of linearized equations which describe the relative motion of spacecraft with respect to a circular reference orbit [11].

Consider two spacecrafts, which we call the *chief* and the *deputy*, orbiting a planet. Note here that the chief satellite does not need to be real, it is often simply a fictitious satellite with an uncontrolled, unperturbed dynamics which defines a reference orbit. We assume the chief is following a circular orbit around the planet and that its motion is unforced and uncontrolled. If  $\vec{r}_H = [x \ y \ z]^T$  is the deputy position with respect to the chief expressed in LVLH frame and  $n = \sqrt{\mu/a^3}$  is the mean motion of the reference orbit outlined by the chief (which, for a circular orbit, corresponds to the angular rate of the orbit), then the HCW equations are

$$\ddot{z} - 2n\dot{x} - 3n^2z = d_z + u_z, \quad (2.30a)$$

$$\ddot{x} + 2n\dot{z} = d_x + u_x, \quad (2.30b)$$

$$\ddot{y} + n^2y = d_y + u_y, \quad (2.30c)$$

where  $[d_x \ d_y \ d_z]^T$  and  $[u_x \ u_y \ u_z]^T$  are the vectors of orbital perturbations and control accelerations, expressed in the LVLH frame.

Since these equations will be used in control design, it is convenient to write them in state space form. Choosing  $\mathbf{x} = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z}]^T$  as the state vector, where  $\vec{r}_H = [x \ y \ z]^T$  is the relative position of the deputy with respect to the chief in LVLH frame, Eqs. (2.37) take the form

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \quad (2.31)$$

where

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -2n \\ 0 & -n^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3n^2 & 2n & 0 & 0 \end{bmatrix}, \quad (2.32)$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.33)$$

The unforced solution of the HCW equations is given in [11] in terms of the state transition matrix,  $e^{At}$ ,

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0), \quad (2.34)$$

where

$$e^{At} = \begin{bmatrix} 1 & 0 & -6nt + 6\sin(nt) & \frac{4\sin(nt)}{n} - 3t & 0 & -\frac{2}{n} + \frac{2\cos(nt)}{n} \\ 0 & \cos(nt) & 0 & 0 & \frac{\sin(nt)}{n} & 0 \\ 0 & 0 & 4 - 3\cos(nt) & \frac{2}{n} - \frac{2\cos(nt)}{n} & 0 & \frac{\sin(nt)}{n} \\ 0 & 0 & -6n + 6n\cos(nt) & -3 + 4\cos(nt) & 0 & -\sin(nt) \\ 0 & -n\sin(nt) & 0 & 0 & \cos(nt) & 0 \\ 0 & 0 & 3n\sin(nt) & 2\sin(nt) & 0 & \cos(nt) \end{bmatrix}, \quad (2.35)$$

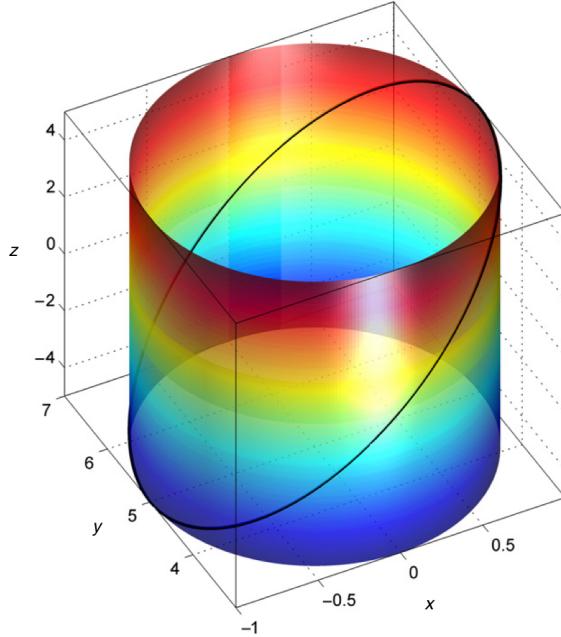


Figure 2.5: 3D ellipse generated by a section of the 3D cylinder when  $\alpha_y = \alpha_z$  [11].

with the initial conditions  $\mathbf{x}(0) = [x(0) \quad y(0) \quad z(0) \quad \dot{x}(0) \quad \dot{y}(0) \quad \dot{z}(0)]^T$ .

#### 2.4.1 The Projected Circular Orbit

A worth mentioning characteristic of this set of equations regards the geometric interpretation that applies when some relation on the initial conditions is satisfied.

First, as pointed out in [11], the HCW equations decouples the out-of-plane motion (along y) from the in-plane motion (xz-plane). Moreover, the *along-track* component of the in-plane motion (which is, the x component) presents some drift, which renders the motion unstable. It can be removed by choosing appropriate initial conditions, and in particular by enforcing the no-drift condition

$$\dot{x}(0) = -2nz(0). \quad (2.36)$$

Now, if (2.36) is satisfied, the solutions to the relative position components can be written in the magnitude-phase form, which constitute a parametric representation of an elliptic cylinder, as shown in Fig. 2.5,

$$x(t) = \rho_x + 2\rho_z \cos(nt + \alpha_z), \quad (2.37a)$$

$$y(t) = \rho_y \sin(nt + \alpha_y), \quad (2.37b)$$

$$z(t) = \rho_z \sin(nt + \alpha_z), \quad (2.37c)$$

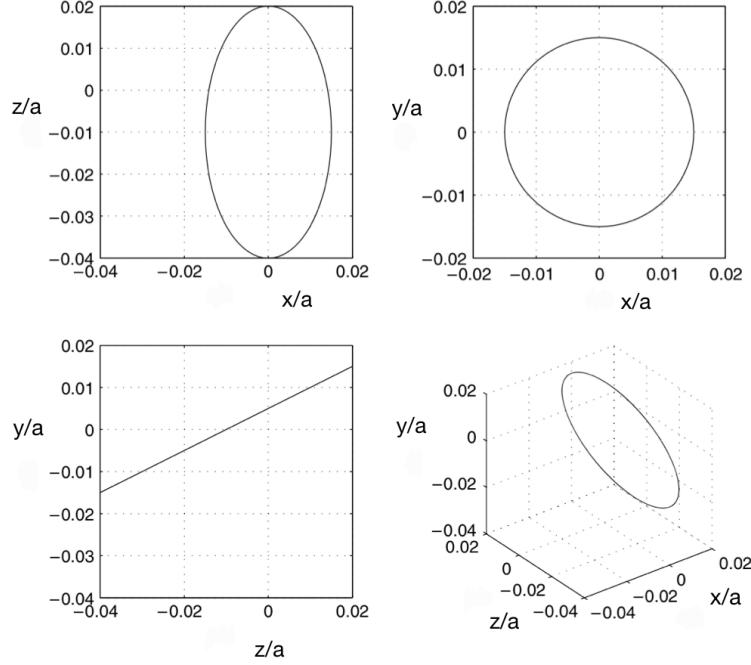


Figure 2.6: Geometry of a relative orbit using the HCW equations for relative motion, with circular xy-projection and linear zy-projection [11].

where the coefficients are given by

$$\rho_x = x(0) - \frac{2\dot{z}(0)}{n}, \quad (2.38)$$

$$\rho_y = \frac{\sqrt{\dot{y}^2(0) + y^2(0)n^2}}{n}, \quad (2.39)$$

$$\rho_z = \frac{\sqrt{\dot{z}^2(0) + z^2(0)n^2}}{n}, \quad (2.40)$$

$$\alpha_y = \tan^{-1} \left( \frac{ny(0)}{\dot{y}(0)} \right), \quad (2.41)$$

$$\alpha_z = \tan^{-1} \left( \frac{n z(0)}{\dot{z}(0)} \right). \quad (2.42)$$

By appropriately choosing these parameters, one can design the relative orbit between two space-crafts and, in turn, the full formation. In particular, the parameters  $\alpha_y$  and  $\alpha_z$  are used to specify the angular offsets between satellites. For instance, setting

$$\alpha_z = \alpha_y \quad , \quad \rho_y = 2\rho_z \quad (2.43)$$

results in the well-studied *projected circular orbit* (PCO), with a linear zy-projection and a circular xy-projection [11], as depicted in Fig. 2.6.

Note that this kinds of trajectories follow from selecting appropriate initial conditions for (2.34), which is the solution of the unforced HCW equations. This means that they are sort of “natural motion trajectories”, spontaneously followed in the disturbance free case. The fact that they do not need thrust is very important as fuel efficiency is a critical feature in space.



# Chapter 3

## Controller Design

Now that we derived the equations governing the motion of the satellites in the formation, we need a way to (locally) control them, such that they correctly and efficiently follow the orbit enforced by the higher-level controller. The goal of this chapter is to present the mathematics used to control the state of a spacecraft in the formation.

### 3.1 The Linear Quadratic Regulator

Each spacecraft in the formation implements a local Linear Quadratic Regulator (LQR), an optimal controller acting on the linear system dynamics (2.31).

Consider the single satellite. When the upper level formation controller sets the reference orbit  $\bar{x}$  that the spacecraft must track, the lower level controller receives it as a setpoint and computes a control input  $u = u(e)$ , function of the tracking error  $e = x - \bar{x}$ .

In particular, the LQR determines the optimal control law  $u = u(e)$  to minimize the following quadratic cost function

$$J(u) = \int_0^\infty (e^T Q e + u^T R u) \, dx \quad (3.1)$$

subject to the linear continuous-time system dynamics (2.31).

In (3.1),  $Q \geq 0$  and  $R > 0$  are, respectively, the state and control weight matrices, and they are selected to weight the relative importance of the performance measures caused by the state vector  $x$  and the control vector  $u$  [20].

The resulting feedback control law is

$$u(t) = -R^{-1}B^T P e = -K e \quad (3.2)$$

where  $K = -R^{-1}B^T P$  and  $P$  is the (constant) solution of the *algebraic Riccati equation* (ARE)

$$0 = PA + A^T P - PBR^{-1}B^T P + Q. \quad (3.3)$$

### 3.2 Choice of LQR Weights and Stability Analysis

The simplest choice for the weight matrices is to set  $Q = I$  and  $R = \kappa I$ , where  $I$  is an identity matrix of appropriate dimension. Here the parameter  $\kappa$  is varied in a sort of trial and error attempt in order to get a good response.

Another way to select the weights is to set  $Q$  and  $R$  to be diagonal matrices with different

diagonal weights:

$$Q = \begin{bmatrix} q_1 & & \\ & \ddots & \\ & & q_n \end{bmatrix}, \quad (3.4)$$

$$R = \kappa \begin{bmatrix} r_1 & & \\ & \ddots & \\ & & r_n \end{bmatrix}. \quad (3.5)$$

Then, each  $q_i$  and  $r_i$  is chosen based on the max error we want to allow, as in [17]. So, if  $x_1$  represents distance in meters, as in our case, and it is okay for us to have an error of 100 m, then we will set  $q_1 = (\frac{1}{100})^2$ , such that  $q_1 x_1^2 = 1$  when  $x_1 = 100$  m.

Let us now analyse briefly stability of the closed-loop system. With  $A$  and  $B$  given by Eqs. (2.32) and (2.33), the pair  $(A, B)$  is controllable and, if we select  $Q$  using one of the two methods above, the pair  $(A, Q^{\frac{1}{2}})$  is observable, so the solution of the ARE is positive definite [11]. This guarantees that the system (2.31) with control law (3.2)

$$\dot{\mathbf{x}} = (A - BK)\mathbf{x} \quad (3.6)$$

is asymptotically stable.

## Chapter 4

# Coverage Coordination Game

In order to implement FES, i.e., close the loop between a cluster of locally controlled satellites and an equilibrium-seeking algorithm, we first need to model the satellites as agents in a (generalized) noncooperative game, whose solutions, i.e., the (generalized) Nash Equilibria, correspond to the desired formation shape. In this chapter, we design two games to model the acquisition (i.e. establishing the formation) and the formation-keeping problems. The decision of distinguishing between the two problems is motivated by the fact that formation-keeping requires milder assumptions on the game setup. However, it can be also achieved with the game designed for the acquisition problem. Finally, we present an equilibrium-seeking algorithms to solve them in a distributed fashion.

### 4.1 Game Design

Consider the coverage coordination problem where we want to “optimally” allocate a set of  $N$  satellites across a circular orbit around a (possibly fictitious) chief orbiting Earth. In particular, we choose a formation where the spacecrafsts must be equally-spaced on the orbit.

Each satellite  $i \in \mathcal{I}$  is modeled as an agent (i.e., a player), where  $\mathcal{I} := \{1, \dots, N\}$  is the set of all players.

For all  $i \in \mathcal{I}$ , the action space  $\Omega_i \in \mathbb{R}$  of player  $i$  is defined as the set of physical locations on the orbit that the satellite can reach. If we specify these locations as angular positions on the orbit, we have  $\Omega_i = [0, 2\pi]$ ,  $\forall i = 1, \dots, N$ . Each player needs to select an action  $x_i \in \Omega_i$  among the available ones.

The exchange of information among the agents is described by a graph  $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ , where  $\mathcal{I}$  is the set of nodes (i.e., of players) and  $\mathcal{E} \subset \mathcal{I} \times \mathcal{I}$  is the set of edges. The unordered pair  $(j, i)$  is in  $\mathcal{E}$  if agent  $i$  can communicate with agent  $j$ , and viceversa. In this case, we say that agent  $j$  is a neighbor of agent  $i$ , and belongs to agent  $i$ 's neighbor set  $\mathcal{N}_i := \{j | (j, i) \in \mathcal{E}\}$ . We assume that the communication between agents is bidirectional, so the resulting graph is undirected.

As pointed out above, the global objective is to optimally cover the circular orbit. From the point of view of the single agents, this can be reformulated as the individual objective of maximizing distance between neighbors. Therefore, we assign to each player a utility function, to be maximized, of the form:

$$U_i(x_i, x_{-i}) = c \sum_{j \in \mathcal{N}_i} d(x_i, x_j) \quad (4.1)$$

where  $x_{-i} := [x_j]_{j \in \mathcal{I} \setminus \{i\}}$  is the strategy profile of all the players except  $i$ ,  $c \in \mathbb{R}$  is a constant and  $d(x_i, x_j)$  is an appropriate distance function representing angular distance between players  $i$  and  $j$ .

*j.* The metric used to specify distance between the agents should be selected while accounting for the following aspects:

- i. the players are moving in a circle. As so, we need a metric that is minimized when, for example, player  $i$  is in 0, i.e.,  $x_i = 0$ , and player  $j$  is in  $2\pi$ , i.e.,  $x_j = 2\pi$ , and maximized when, for instance, player  $i$  is in 0, i.e.,  $x_i = 0$ , and player  $j$  is in  $\pi$ , i.e.,  $x_j = \pi$ ;
- ii.  $d(\cdot, \cdot)$  should have a meaningful interpretation also with respect to the physical coordinates on the orbital plane, i.e., the position  $(y, z)$  of the satellites.

With this in mind, without loss of generality, let us consider the orbit to be the unit circle and, thus, set  $y = \cos(\theta)$ ,  $z = \sin(\theta)$ , with  $\theta$  being the angular position on the unit circle. The distance between two agents located at  $(y, z)|_{\theta_i}$  and  $(y, z)|_{\theta_j}$  respectively, can be found to be

$$d(\theta_i, \theta_j) = \sqrt{2 - 2 \cos(\theta_i - \theta_j)}. \quad (4.2)$$

Using utility function (4.1), with distance defined as in (4.2), gives us  $N$  inter-dependent optimization problems whose collection constitutes a game:

$$\begin{aligned} \forall i \in \mathcal{I} : \quad & \min_{x_i \in \mathbb{R}} \quad -U(x_i, x_{-i}) \\ & s.t. \quad x_i \in \Omega_i \end{aligned} \quad (4.3)$$

Next, we present in details the differences between the acquisition and formation-keeping games.

#### 4.1.1 Acquisition Game

The acquisition game models the problem of acquisition of the formation.

Here, each player  $i \in \mathcal{I}$  wants to maximize the distance from its neighbors while maintaining a maximum distance between at least two of them (the right and left neighbors in the circular orbit). This assumption is motivated by the fact that, in a real setting, it is necessary to keep a certain degree of connectivity between communicating spacecrafts. To account for this, we impose the following coupling constraint

$$|Ax| \leq b, \quad (4.4)$$

where  $A \in \mathbb{R}^{|\mathcal{E}| \times N}$  is the (transpose) incidence matrix of a graph  $\mathcal{G} = (\mathcal{I}, \mathcal{E})$  to be defined,  $x = [x_i]_{i \in \mathcal{I}} \in \Omega \subset \mathbb{R}^N$  denotes the stacked vector of all the agents' decisions, and  $b = d_{max} \mathbb{1}_{|\mathcal{E}|}$ , with  $d_{max} \in \mathbb{R}_+$  determining the maximum allowed distance between two neighbors, and  $\mathbb{1}_{|\mathcal{E}|}$  being a column vector of  $|\mathcal{E}|$  ones. In particular, the graph  $\mathcal{G}$  will have a ring topology, such that each player only has two neighbors.

The presence of the coupling constraints influences the game mainly in two ways. First, it limits the decisions that player  $i$  can take, because not all the elements of player  $i$ 's action set  $\Omega_i$  will satisfy the constraint. For this reason, for each player  $i \in \mathcal{I}$ , we define a feasible decision set  $X_i(x_{-i})$ , function of  $i$ 's neighbors' position, which accounts for (4.4), i.e.,

$$X_i(x_{-i}) = \Omega_i \cap \{x_i \in \mathbb{R} \mid |A[x_i; x_{-i}]| \leq b\} \quad (4.5)$$

Second, it requires that all the agents in some sense agree with the constraint. To this end, we introduce a local multiplier  $\lambda_i \in \mathbb{R}_+^m$  and a local auxiliary variable  $z_i \in \mathbb{R}^m$ , used for the coordination needed to satisfy the coupling constraint and to reach consensus on the local multipliers  $\lambda_i$ . When the game is solved, ideally players should have reached consensus on the value of the multiplier.

To model information sharing between players, we define two graphs, as in [23]:

- $\mathcal{G}_f = (\mathcal{N}, \mathcal{E}_f)$  is the *interference graph*. This graph is used by the players to exchange their local position  $x_i$  and it determines their utility functions. We assume that players are equipped with a sensor that allows them to sense the presence of other agents within a fixed range. If player  $j$  finds himself sufficiently close to player  $i$ , then  $(j, i) \in \mathcal{E}_f$ , we say that player  $j$  is an *interference neighbor* of player  $i$ , and  $j \in \mathcal{N}_i^f = \{j | (j, i) \in \mathcal{E}_f\}$ .
- $\mathcal{G}_\lambda = (\mathcal{N}, \mathcal{E}_\lambda)$  is the *multiplier graph*, through which players exchange local  $\{\lambda_i, z_i\}$ . It is a fixed, ring graph, and so the *multiplier neighborhood* of player  $i$ ,  $\mathcal{N}_i^\lambda = \{j | (j, i) \in \mathcal{E}_\lambda\}$ , only contains two agents, from which  $i$  can receive (and to which it can send) local information regarding the constraint. The neighbors defined by  $\mathcal{G}_\lambda$  are also the ones considered by each  $i$  in the constraint (4.4). Note that, by exploiting a ring topology,  $\mathcal{G}_\lambda$  satisfies Assumption 3 of [23], namely, it is undirected, connected and symmetric.

#### 4.1.2 Formation-keeping Game

Once the desired formation has been reached, it must be kept despite the presence of disturbances, namely, the atmospheric drag, the moon's gravity and the  $J_2$  perturbation. The formation-keeping game helps us with this.

The setup for this game is simpler than the one considered for the acquisition game. No constraint is needed to guarantee correct maintenance of the formation, and so the only local information of player  $i$  regard its feasible set  $\Omega_i$  and its local decision  $x_i \in \mathbb{R}$ .

The position information is bidirectionally shared through an *interference graph*  $\mathcal{G}_f = (\mathcal{N}, \mathcal{E}_f)$  defined as above.

## 4.2 Game Solution

To solve the games, we use the distributed algorithm for computation of generalized Nash Equilibria in noncooperative games proposed in Algorithm 1 in [23].

First, note that the utility function of each player  $i \in \mathcal{I}$ , (4.1), with distance defined as in (4.2), is differentiable and convex with respect to  $x_i$  given any fixed  $x_{-i}$ , and  $\Omega_i$  is a closed and convex set. Moreover,  $X = \prod_{i=1}^N X_i$  and  $X_i(x_{-i})$ , given any fixed  $x_{-i}$ , have a nonempty interior. The algorithm is adapted and reported here for clarity.

**Algorithm 1.** Preconditioned forward-backward method

*Initialization:*  $x_{i,0} \in \Omega_i$ ,  $\lambda_{i,0} \in \mathbb{R}_+^m$ , and  $z_{i,0} \in \mathbb{R}^m$ .

*Iteration:* For each player  $i$

- Step 1: Receives  $x_{j,k}$ ,  $j \in \mathcal{N}_i^f$ ,  $\lambda_{j,k}$ ,  $j \in \mathcal{N}_i^\lambda$  and updates

$$x_{i,k+1} = P_{\Omega_i} [x_{i,k} - \tau_i (\nabla_{x_i} (-U_i(x_{i,k}, x_{-i,k})) - A_i^T \lambda_{i,k})], \\ z_{i,k+1} = z_{i,k} + \nu_i \sum_{j \in \mathcal{N}_i^\lambda} w_{ij} (\lambda_{i,k} - \lambda_{j,k}).$$

- Step 2: Receives  $z_{j,k+1}$ ,  $j \in \mathcal{N}_i^\lambda$  and updates

$$\lambda_{i,k+1} = P_{\mathbb{R}_+^m} \{ \lambda_{i,k} - \sigma_i [A_i (2x_{i,k+1} - x_{i,k}) - b_i \\ + \sum_{j \in \mathcal{N}_i^\lambda} w_{ij} [2(z_{i,k+1} - z_{j,k+1}) - (z_{i,k} - z_{j,k})] \\ + \sum_{j \in \mathcal{N}_i^\lambda} w_{ij} (\lambda_{i,k} - \lambda_{j,k})] \}$$

---

where  $\nabla_{x_i} (-U_i(x_{i,k}, x_{-i,k}))$  is the gradient, with respect to  $x_i$ , of player  $i$ 's utility function (4.1),  $x_{i,k}$ ,  $z_{i,k}$  and  $\lambda_{i,k}$  denote  $x_i$ ,  $z_i$  and  $\lambda_i$  at iteration  $k$ ,  $\tau_i$ ,  $\nu_i$  and  $\sigma_i$  are fixed constant step-sizes of player  $i$  and  $W = [w_{ij}]$  is the adjacency matrix of  $\mathcal{G}_\lambda$ . Moreover,  $A_i$  and  $b_i$  represent the blocks of the constraint matrices  $A$  and  $b$  in (4.4) that regard directly agent  $i$ , and that agent  $i$  knows locally.

For the non-generalized formation-keeping game, we can use a simplified version of Algorithm 1, where the dual variables are disregarded:

---

**Algorithm 2.** Projected-pseudo-gradient method

---

*Initialization:*  $x_{i,0} \in \Omega_i$ .

*Iteration:* Player  $i$  receives  $x_{j,k}$ ,  $j \in \mathcal{N}_i^f$  and updates

$$x_{i,k+1} = P_{\Omega_i} [x_{i,k} - \tau_i \nabla_{x_i} f_i(x_{i,k}, x_{-i,k})].$$


---

As pointed out in [23], the distributed nature of Algorithms 1 and 2 follows from the facts that (i) each player only needs to know its local data (local objective function, local feasible set, local constraint, local multiplier and auxiliary variable), and (ii) there is no centralized coordinator to update and broadcast a common multiplier, while instead consensus on it is reached thanks to local communication through  $\mathcal{G}_\lambda$  only.

# Chapter 5

# Feedback Equilibrium Seeking Control

Feedback equilibrium seeking (FES) is an extension of feedback optimization [8] where the objective is to design a feedback controller that drives a physical plant to the solution trajectory of a time-varying generalized equation (GE) [2]. In particular, GEs can be used to model (generalized) Nash equilibria of noncooperative games. We start this chapter by briefly presenting the concept of FES, in its generality, and then tailor this general framework to the formation control problem described in the previous chapters.

## 5.1 General Design Framework

The following description of FES is taken from [2]. For a more detailed explanation, we refer to the paper.

### 5.1.1 Problem Setting

In feedback equilibrium seeking, the control of a non-linear continuous-time state space system

$$\dot{x}(t) = f(x(t), u(t), w(t)), \quad (5.1a)$$

$$y(t) = g(x(t), w(t)), \quad (5.1b)$$

is addressed, where  $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n_x}$  is the state,  $u : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n_u}$  is the control input,  $w : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n_w}$  is an exogenous disturbance, and  $y : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n_y}$  is the output of the system. The function  $f$  is assumed to be locally Lipschitz continuous and other assumptions regarding continuity of  $g$  and compactness and convexity of  $\mathcal{U} \subset \mathbb{R}^{n_u}$ , as well as compactness of  $\mathcal{W} \subset \mathbb{R}^{n_w}$ , are needed.

The existence of a continuously differentiable steady-state mapping  $x_{ss} : \mathcal{U} \times \mathcal{W} \rightarrow \mathbb{R}^{n_x}$  such that  $\forall u \in \mathcal{U}, w \in \mathcal{W}, f(x_{ss}(u, w), u, w) = 0$ , for the system in (5.1), allows us to define a steady-state input-output mapping for the system (5.1), denoted by

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

Moreover, the output  $y$  of the system is assumed to be measured, while the exogenous disturbance  $w$  is unmeasured.

The goal is to design a feedback controller to steer (5.1) to an economic equilibrium, which coincides with the solution of the following parameterized GE:

$$0 \in F(u, y) + \mathcal{B}(u), \quad (5.3a)$$

$$y = h(u, w), \quad (5.3b)$$

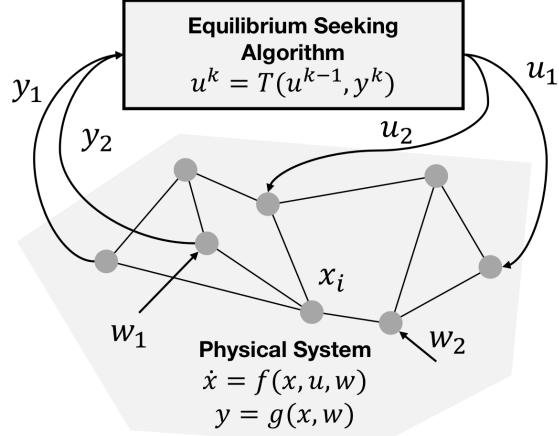


Figure 5.1: Measurements from a physical system are fed into an equilibrium seeking algorithm [2].

where  $F : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_u}$  is continuously differentiable and  $\mathcal{B} : \mathcal{U} \rightrightarrows \mathbb{R}^{n_u}$  is a set-valued mapping. The mathematical formalization of the problem, taken from [2], can be stated as: design an output feedback controller that will drive (5.1) to solution trajectories  $y^*(w(t))$  and  $u^*(w(t))$  of (5.3) where

$$\begin{bmatrix} u^*(w) \\ y^*(w) \end{bmatrix} = \{u \in \mathcal{U}, y \in \mathbb{R}^{n_y} \mid F(u, y) + \mathcal{B}(u) \ni 0, \quad y = h(u, w)\} \quad (5.4)$$

is the solution-to-parameter mapping, with  $u^*$  and  $y^*$  single-valued functions.

Note that encoding the desired “efficient” operating conditions using GEs allows us to formulate under this framework, among others, (constrained) optimization problems and (generalized) Nash games.

### 5.1.2 Control Strategy

Figure 5.1 shows the general idea behind FES. A loop is closed around a physical plant described by Eqs. (5.1) and an equilibrium seeking algorithm, in order to make the system work near the economic operating conditions defined in (5.4). Measurements of the output  $y$  obtained from the system are fed to an equilibrium seeking algorithm, which in turns acts as a controller, providing control actions  $u$  to the system. The reason why the control input is computed numerically in this manner is twofold: (i) the exogenous disturbance  $w$  is unmeasurable, so we can not simply pick  $u$  as a function of it, i.e.,  $u(t) = u^*(w(t))$ ; (ii) often, the solution  $u^*$  is not available in closed form, and so numerical computation is needed.

In order to solve (5.3), a discrete algorithm is considered

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

where  $T : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_u}$  is how the next iterate is generated.

As pointed out above, with  $w$  being not measurable, exact evaluations of the steady-state map  $h$  are not possible, and these are replaced with measurements of  $y$  obtained from the system. Since the algorithm is discrete, these measurements are sampled with sampling period  $\tau > 0$ , and the resulting control inputs  $u$  are fed continuously to the plant through a zero-order hold. Denoting with  $t^k = k\tau$  the sampling instants, the resulting sampled-data closed-loop system is

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

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

where  $\epsilon \in (0, 1]$  is a parameter used to weight the control action computed by the algorithm (5.5).

## 5.2 Satellite Formation Control using FES

In this section, the general framework of FES described in section 5.1 is specialized for the formation control problem. In particular, we will consider the acquisition task, but the following analysis can be adapted also to the formation-keeping task.

Recall that what we want to achieve is equally-spaced allocation of a group of  $N$  satellites on a circular orbit around a (fictitious) chief. We assume the target orbit corresponds to the orbit where the satellites are deployed, so the upper level formation controller (see Fig. 1.1) has to take care only of the spacing between spacecrafts.

In section 2.4.1, we showed how the formation can be designed through the choice of the parameters  $\rho_x$ ,  $\rho_y$ ,  $\rho_z$ ,  $\alpha_y$  and  $\alpha_z$ , and how, in particular,  $\alpha_y \in \mathbb{R}^N$  and  $\alpha_z \in \mathbb{R}^N$  determine the relative angular offsets of the  $N$  satellites. Moreover, since we want to get the satellites in a circular orbit, following condition (2.43) in section 2.4.1, we set

$$\alpha_{y,i} = \alpha_{z,i}, \quad \forall i \in \mathcal{I}, \quad (5.7)$$

where  $\alpha_{y,i}$  and  $\alpha_{z,i}$  are, respectively, the  $i$ -th element of  $\alpha_y$  and  $\alpha_z$ .

Let us define the input to the system given by the upper level formation controller as  $u_u = \alpha$ ,  $u_u \in \mathcal{U}_u$ , where  $\mathcal{U}_u \subset \mathbb{R}^N$  and  $\alpha = \alpha_y = \alpha_z$ .

Consider now the acquisition game designed in section 4.1, played by  $N$  agents, each choosing an action in the respective feasible sets  $\Omega_i$ ,  $\forall i \in \mathcal{I}$ . In our setting, these actions are the control inputs  $u_{u,i}$ , where  $u_{u,i}$  is the  $i$ -th element of  $u_u$ , and they correspond to

$$u_{u,i} = \alpha_i,$$

where  $\alpha_i$  is the  $i$ -th element of  $\alpha$ .

If we refer to the desired reference orbit for the  $i$ -th satellite as  $\bar{x}_i(t, u_{u,i})$ , the local lower-level stabilizer (see section 3.1) generates an input

$$u_{l,i} = -K(x_i - \bar{x}_i(t, u_{u,i})), \quad (5.8)$$

where  $K$  is the solution of the algebraic Riccati equation (3.3) and  $x_i$  is the state vector of the  $i$ -th spacecraft as defined in section 2.4.

The linear system in state-space form (2.31), with state and control matrices given by (2.32) and (2.33) and control input (5.8), can be written, for the  $i$ -th satellite, as

$$\dot{x}_i(t) = Ax_i(t) + B(u_{l,i}(t) + w_i(t)) = (A - BK)x_i(t) + BK\bar{x}_i(t, u_{u,i}) + Bw_i(t). \quad (5.9)$$

Assuming that the state  $x$  is measurable and setting this as the output of our system, and since the dynamics of the  $N$  agents are decoupled, (5.1) can be expressed by the following linear continuous-time state-space system

$$\forall i \in \mathcal{I} \quad : \quad \begin{cases} \dot{x}_i(t) = (A - BK)x_i(t) + BK\bar{x}_i(t, u_{u,i}) + Bw_i(t), \\ y_i(t) = x_i(t), \end{cases} \quad (5.10)$$

where  $x_i : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^6$  is the state,  $u_{u,i} : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^2$  is the upper level control input,  $w_i : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^3$  is an exogenous disturbance representing the effects of the orbital perturbations (see section 2.3), and  $y_i : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^6$ , is the output of the system.

By setting the state equation of (5.10) to 0, we get the steady-state mappings

$$x_{ss,i} = (A - BK)^{-1} (-BK\bar{x}_i(u_{u,i}) - Bw_i), \quad (5.11)$$

which allows us to write the local steady-state input-output mappings in (5.2) as

$$h_i(u_{u,i}, w_i) = x_{ss,i}(u_{u,i}, w_i) = (A - BK)^{-1} (-BK\bar{x}_i(u_{u,i}) - Bw_i). \quad (5.12)$$

Then, the utility function (4.1) which each agent  $i$  is equipped with, is of the form  $U_i(u_{u,i}, h(u_u, w))$ , being it dependent on agent  $i$ 's action and on the actions of some other agents. Note that  $h(u_u, w)$  is the steady-state input-output mapping of the entire system (5.10).

The maximization of these utility functions performed by each agent results in  $N$  inter-dependent optimization problems, thus constituting a game:

$$\forall i \in \mathcal{I} \quad : \quad \min_{u_{u,i} \in \mathcal{U}_{u,i}} \{-U_{u,i}(u_{u,i}, y) \quad | \quad y = h(u_u, w)\}. \quad (5.13)$$

Such a game is solved by finding a Nash equilibrium, a condition in which no agent can increase its utility by unilaterally changing action. Since the utility function (4.1) with (4.2) is convex and continuously differentiable in  $u_{u,i} \in U_{u,i}$ , for all  $u_{u,-i}$ , by defining the game map  $F = [F_i]_{i \in \mathcal{I}}$ , with components

$$F_i(u_{u,i}, y) := \nabla_{u_{u,i}} U_{u,i}(u_{u,i}, y) + \nabla_{u_{u,i}} h_i(u_{u,i}, w_i)^T \nabla_{y_i} U_{u,i}(u_{u,i}, y), \quad (5.14)$$

the strategy profile  $\bar{u}_u = [\bar{u}_{u,i}]_{i \in \mathcal{I}}$  is a NE of the game if and only if it verifies:

$$0 \in F(\bar{u}_u, h(\bar{u}_u, w)) + \mathcal{N}_{\mathcal{U}_u}(\bar{u}_u), \quad (5.15)$$

where  $\mathcal{U}_u = \prod_{i \in \mathcal{I}} \mathcal{U}_{u,i}$  and  $\mathcal{N}_{\mathcal{U}_u} = \partial \iota_{\mathcal{U}_u}$  is the normal cone operator of the indicator function of  $\mathcal{U}_u$ .

Now that we have established the dynamics of the cluster of satellites (5.10), defined the steady-state mappings (5.12) and described how the GE in (5.3) can be used to encode the acquisition game, we can proceed with the definition of the sampled-data closed-loop system.

In order to do this, what we need to find is a suitable  $T$ , i.e., the rule for generating the next iterate, as defined in (5.5). To this end, Algorithm 1 from section 4.2 can be employed.

The resulting sampled-data closed-loop system is

$$\Sigma_1^s : \forall i \in \mathcal{I} \quad : \quad \begin{cases} \dot{x}_i(t) = (A - BK)x_i(t) + BK\bar{x}_i(t, u_{u,i}) + Bw_i(t), \\ y_i(t) = x_i(t), \end{cases} \quad (5.16a)$$

$$\Sigma_2^s : \forall i \in \mathcal{I} \quad : \quad \begin{cases} u_{u,i}^k = (1 - \epsilon)u_{u,i}^{k-1} \\ \quad + \epsilon P_{\Omega_i} \left[ u_{u,i}^{k-1} - \tau_i \left( \nabla_{u_{u,i}} (-U_{u,i}(u_{u,i}^{k-1}, y)) - A_i^T \lambda_i^{k-1} \right) \right], \\ z_i^{k+1} = z_i^k + \nu_i \sum_{j \in \mathcal{N}_i^\lambda} w_{ij} (\lambda_i^k - \lambda_j^k), \\ \lambda_i^{k+1} = P_{\mathbb{R}_+^m} \{ \lambda_i^k - \sigma_i [A_i (2u_{u,i}^{k+1} - u_{u,i}^k) - b_i \\ \quad + \sum_{j \in \mathcal{N}_i^\lambda} w_{ij} [2(z_i^{k+1} - z_j^{k+1}) - (z_i^k - z_j^k)] \\ \quad + \sum_{j \in \mathcal{N}_i^\lambda} w_{ij} (\lambda_i^k - \lambda_j^k)] \}, \\ u_{u,i}(t) = u_{u,i}^k, \quad \forall t \in [t^k, t^{k+1}), \end{cases} \quad (5.16b)$$

where  $\epsilon \in (0, 1]$  is a parameter used to weight the control action computed by the algorithm, and  $\lambda_i$  and  $z_i$  are, respectively, a local multiplier and a local auxiliary variable used in Algorithm 1. Note here the interdependence of the two subsystems: in  $\Sigma_1^s$ , the reference state  $\bar{x}_i$  is computed using  $u_{u,i}$ , generated by  $\Sigma_2^s$ , and in  $\Sigma_2^s$  the utility function  $U_{u,i}$  depends on the state  $x_i$  and on the measurements  $y_i$  through the distance function defined in (4.2).

For a description of the algorithm and a clearer definition of the symbols  $\lambda_i$ ,  $z_i$ ,  $\tau_i$ ,  $\nu_i$ ,  $\sigma_i$ ,  $A_i$ ,  $w_{ij}$  and  $\mathcal{N}_i^\lambda$ , refer to section 4.2.



# Chapter 6

## Simulation Results

In this chapter, we give some simulation examples to illustrate the main results.

We consider the acquisition and formation-keeping of a circular formation constituted by a group of  $N = 6$  equally-spaced satellites. The satellites are batch deployed by a deployment vehicle into a circular, nearly-equatorial, low-Earth orbit (LEO) around planet Earth, which will represent the orbit followed by a (fictitious) chief, at an altitude of approximately 400 km above the surface. An initial radial velocity impulse de-attaches them from the chief and injects them at the same point of a circular orbit of radius  $R = 150$  m centered at the chief's position, from which they start to spread out to establish the formation, by adjusting their relative angular positions, while orbiting around Earth.

The motion of the spacecrafts is perturbed by the atmospheric drag, which constitutes the primary non-gravitational force acting on satellites in near-Earth orbits, the influence of the moon, and the  $J_2$  effect due to non-sphericity of the Earth. For the satellites' drag coefficient, mass and drag surface area we consider the constant values used in [15]. Since the circular orbit around Earth is nearly equatorial, and the deviation of the motion of the satellites from this is small compared to its radius, we assume the value of the atmospheric density to be constant and equal to  $\rho = 3 \times 10^{-3}$  kg/km<sup>3</sup>, a reasonable value for an altitude of circa 400 km. The unitless  $J_2$  coefficient for the Earth is  $J_2 = 1.083 \times 10^{-3}$ . The Earth's equatorial radius is  $R_e = 6378.1633$  km, and the Earth's gravitational constant is  $\mu = 3.9860 \times 10^5$  km<sup>3</sup>/s<sup>2</sup>. The mass of the moon is  $m_{moon} = 7.342 \times 10^{22}$  kg.

Regarding the local linear quadratic regulators in section 3.1, we select the weight matrices as:

$$Q = 10^{-9}I,$$
$$R = \kappa I$$

with  $\kappa = 0.1$ .

The constant  $c$  in the utility function (4.1) of the coverage coordination games is  $c = \frac{1}{2}$  and we set the maximum distance between neighbors, imposed as a constraint in (4.4), to  $d_{max} = \frac{2pi}{6} = \frac{\pi}{3}$ . The sensing range of the satellites, as defined in section 4.1.1, is fixed such that, once the target formation has been acquired, each satellite only communicates with its right and left neighbors, which is, such that the interference graph  $\mathcal{G}_f$  corresponding with the target formation has a ring topology. For a formation of  $N = 6$  satellites, its value is set to  $range = 0.5$ .

The step-sizes in Algorithm 1, section 4.2, are selected as  $\tau_i = 0.03$ ,  $\nu_i = 0.2$  and  $\sigma_i = 0.03$ ,  $\forall i = 1, \dots, 10$ . Initials  $\lambda_{i,0}$  and  $z_{i,0}$  are set to 0.

The sample time is chosen as the orbital period T.

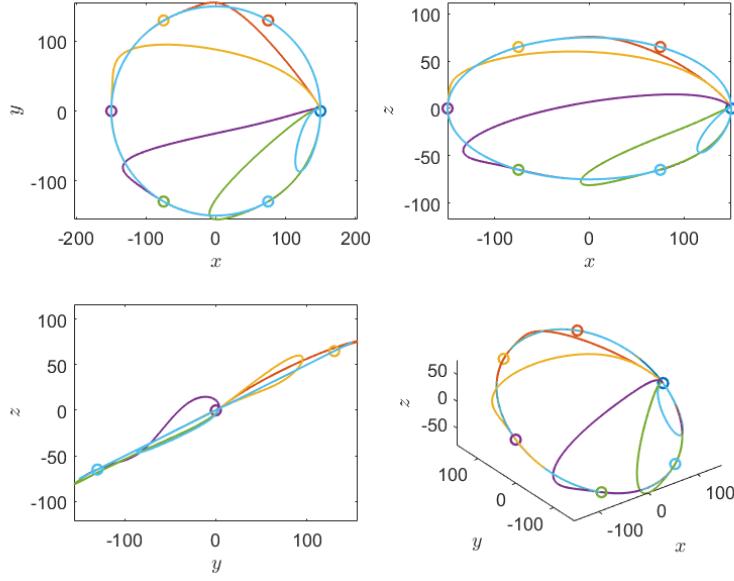


Figure 6.1: PCO in LVLH frame for the open-loop acquisition phase. The satellites, starting from the same initial position, equally-space themselves in the orbit.

#### A. Open-loop acquisition phase

Here we address acquisition of the formation by applying optimal input commands in open-loop, which is, without closing the outer loop in Fig. 1.1. Specifically, we compute the desired spacing among neighboring satellites beforehand, derive the reference orbits  $\bar{x}_i$  for each satellite, and feed them to the locally controlled satellites.

After deployment we assume the following initial conditions for the satellites,  $\forall i$ , expressed in LVLH frame:

$$\begin{aligned} x_{i,0} &= R \cos \phi_{i,0} = 150, \\ z_{i,0} &= \frac{R}{2} \sin \phi_{i,0} = 0, \\ y_{i,0} &= 2n z_{i,0} = 0, \\ \dot{x}_{i,0} &= -2n z_{i,0} = 0, \\ \dot{z}_{i,0} &= nx_{i,0}/2 = 0.0849, \\ \dot{y}_{i,0} &= nx_{i,0} = 0.1697, \end{aligned}$$

where  $n = 0.0011$  is the mean motion of the chief's orbit and  $\phi_{i,0}$  is the initial phase of the  $i$ -th satellite on the circular orbit about the chief, which we set to 0. These initial conditions are selected in order to satisfy relations (2.43) and design a projected circular orbit.

The target orbits for each satellite are computed using (2.37). In particular, the parameters  $\rho_x$ ,  $\rho_y$  and  $\rho_z$  can be calculate with Eqs. (2.38)-(2.40) and are the same for each satellite; the angular offset parameters instead are specific for each spacecraft and they are calculated as:

$$\begin{aligned} \alpha_{y,i} &= \tan^{-1} \left( \frac{ny(0)}{\dot{y}(0)} \right) + \frac{2\pi}{6}(i-1) = \frac{2\pi}{6}(i-1), \\ \alpha_{z,i} &= \alpha_{y,i}, \end{aligned}$$

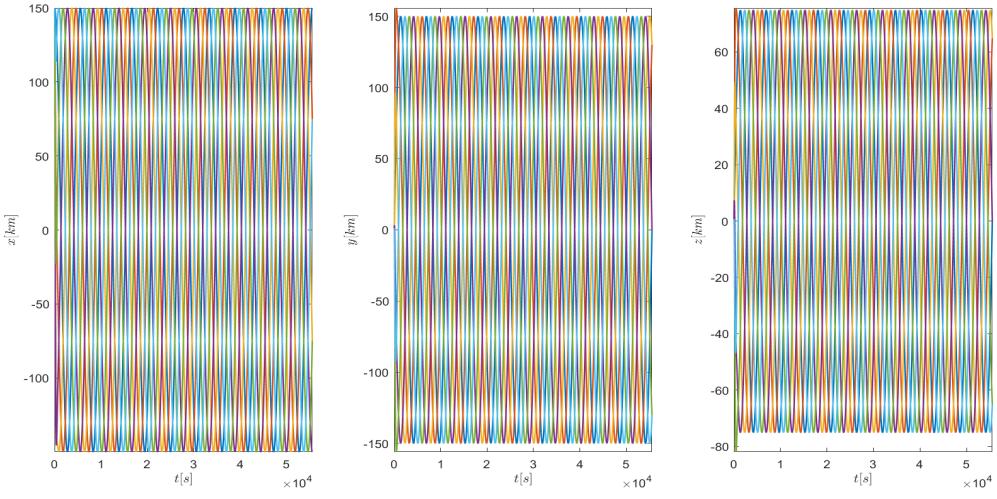


Figure 6.2: Motion along the  $x$ ,  $y$  and  $z$  axes of the LVLH frame during the open-loop acquisition phase.

for  $i = 1, \dots, N$ , where  $\frac{2\pi}{6} = \frac{\pi}{3}$  is the desired spacing between neighbors for a group of  $N = 6$  satellites. The simulation is run for 5 orbits around Earth, each having an orbital period of  $T = 5.5536 \times 10^3$  s.

Figure 6.1 shows the PCO in LVLH frame for the open-loop acquisition phase, while in figure 6.2 the motion of the satellites along the axes of the LVLH frame are depicted.

The local control inputs from the LQRs are showed in Fig. 6.3: after an initial spike, necessary to place the satellites in the desired relative angular positions, the control inputs follow a sinusoidal motion, keeping the spacecrafats in (relative) place while rejecting the disturbances.

#### B. Open-loop formation-keeping phase

At this point, the target formation has been acquired and the satellites must maintain it. We run the simulation in open-loop applying the same optimal input commands computed for the acquisition phase, and so the same reference orbits  $\bar{x}_i$ ,  $\forall i$ . We initialize the state of the satellites using the final states reached at the end of the acquisition phase, which represent the absolute LVLH positions and velocities corresponding to the desired relative states between spacecrafats.

The PCO for the open-loop formation-keeping phase is depicted in Figure 6.5, where we see how the satellites remain on the orbit they are initialized on. Figure 6.6 displays the (local) control inputs necessary to maintain the relative geometry.

#### C. Closed-loop acquisition phase

Unfortunately, because of time constraints, this section was not properly addressed. In particular, the static game for the acquisition phase has been correctly designed and solved, using Algorithm 1 from section 4.2, but it has not been implemented in closed-loop with the physical system. This task is left for future work.

Figures 6.7 and 6.8 shows the solution of the static acquisition game for a formation of  $N = 6$  satellites. We see how the decisions of the players correctly converge to the generalized Nash Equilibrium.

#### D. Closed-loop formation-keeping phase

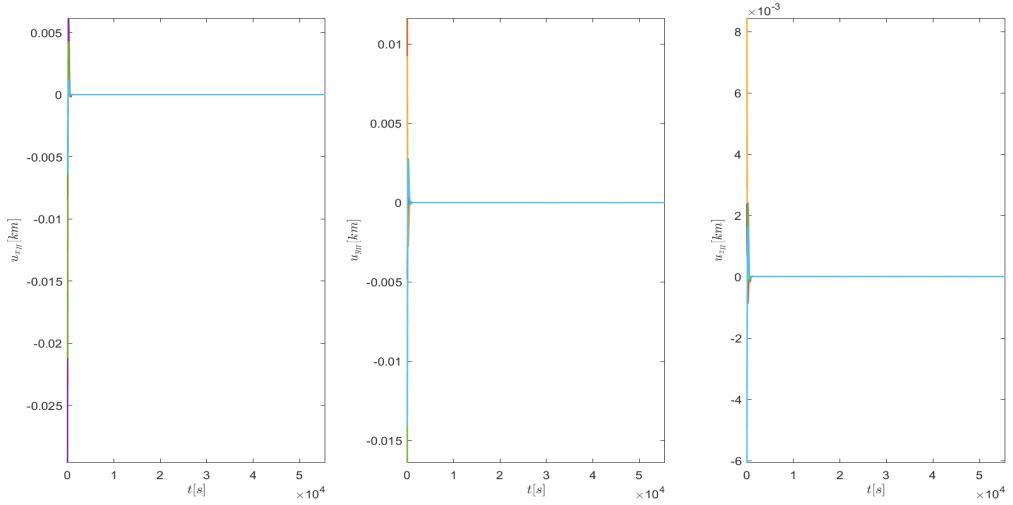


Figure 6.3: Control inputs from the lower-level linear quadratic regulators during the open-loop acquisition of the formation, in LVLH frame.

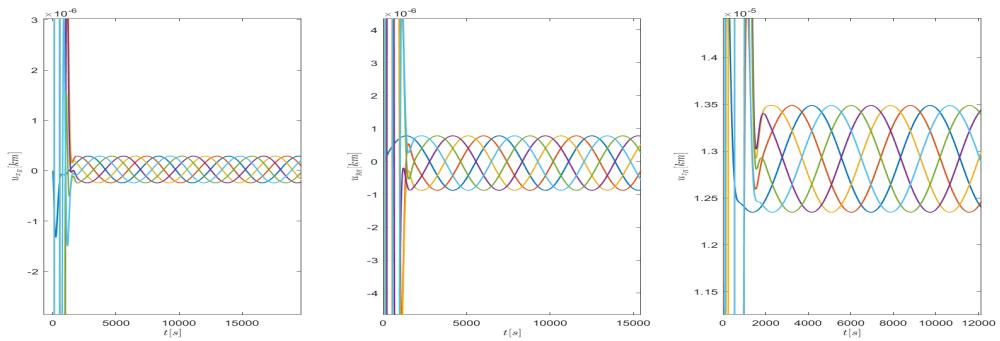


Figure 6.4: Close-up of Fig. 6.3 showing the stabilizing control inputs.

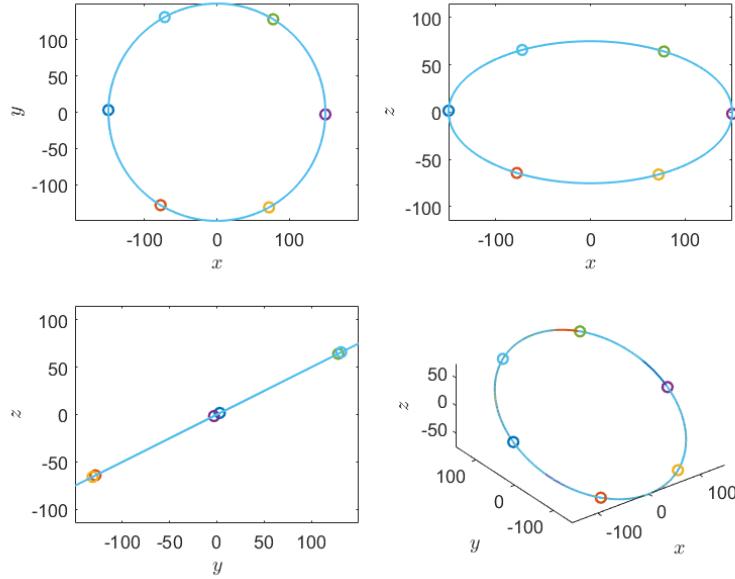


Figure 6.5: PCO representation in LVLH frame for the open-loop formation-keeping phase. The initial orbit is maintained by the satellites.

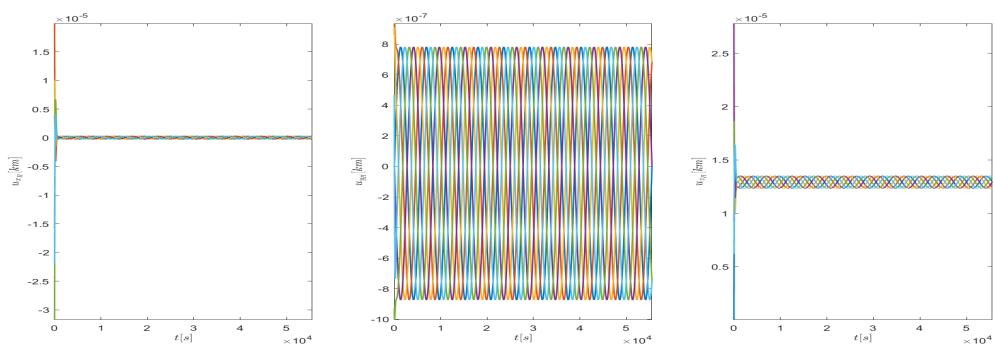


Figure 6.6: Control inputs from the lower-level linear quadratic regulators during the open-loop formation-keeping phase, in LVLH frame.

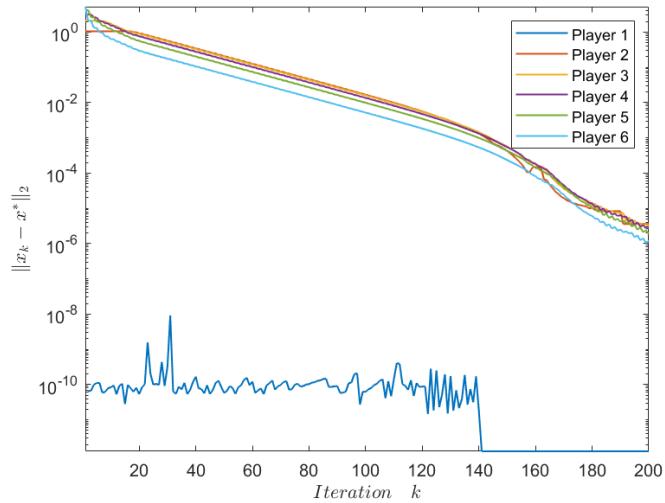


Figure 6.7: Trajectories of  $\|x_k - x^*\|_2$ . The players find the unique generalized Nash Equilibrium.

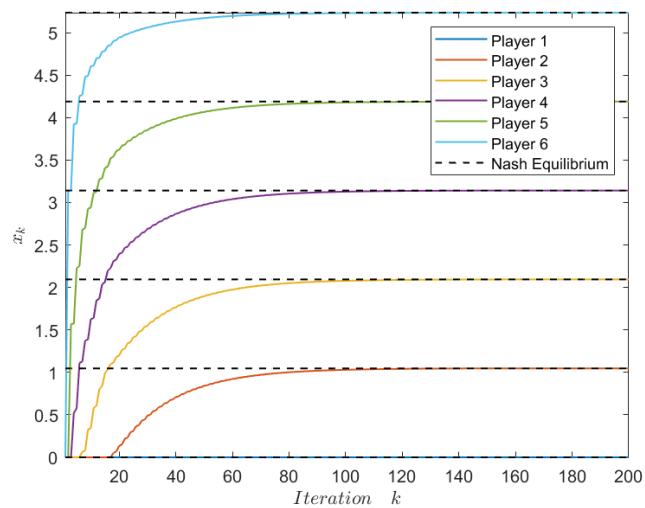


Figure 6.8: Trajectories of the decisions of the players converge to the generalized Nash Equilibrium.

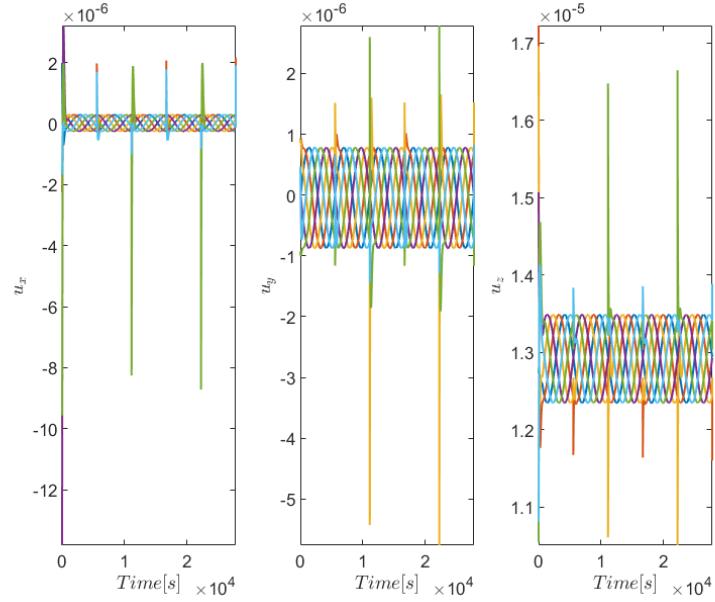


Figure 6.9: Control inputs from the lower-level linear quadratic regulators for the closed-loop formation-keeping phase, in LVLH frame.

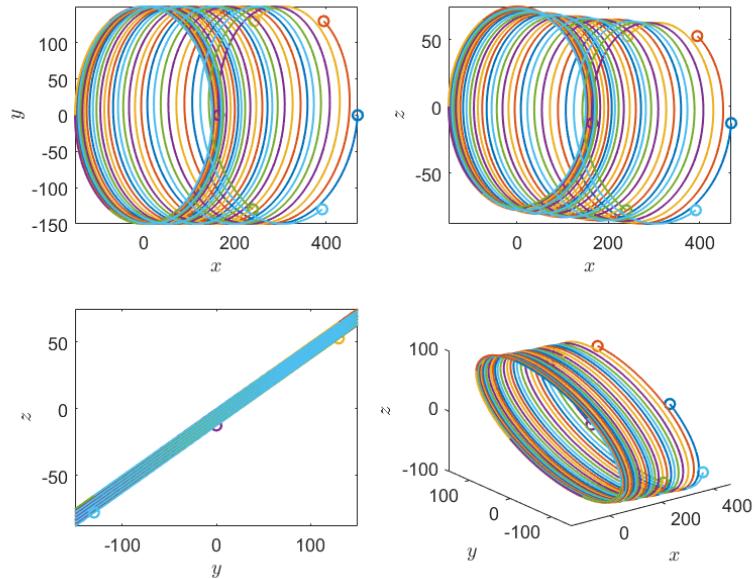


Figure 6.10: Effect of the atmospheric drag on the uncontrolled closed-loop formation-keeping phase.

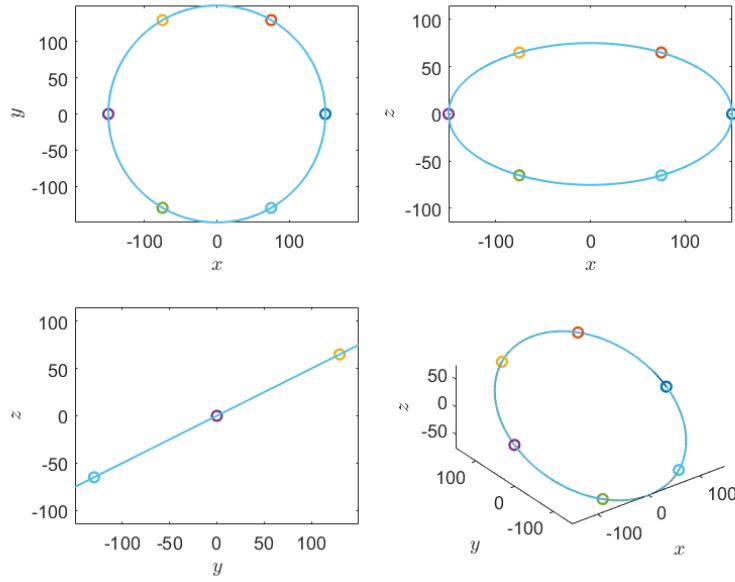


Figure 6.11: PCO in LVLH frame for the controlled closed-loop formation-keeping phase.

We address the same problem as in point *B*, but employing the full control architecture of figure 1.1. Here the loop between the physical system, constituted by the 6 locally controlled satellites, and the equilibrium-seeking algorithm (Algorithm 2 in section 4.2) is closed, and FES is implemented.

Figures 6.10 and 6.11 depict the action of the controller under the influence of the atmospheric drag. We see how the effect of the drag is to produce a drift on the orbit, and how, in the controlled case, the initial orbit is maintained and the satellites correctly hold their relative angular positions. The lower-level stabilizing inputs are presented in figure 6.9.

# Bibliography

- [1] E. Sin M. Arcak and A. Packard. “Small satellite constellation separation using linear programming based differential drag commands”. In: *Proceedings of the 2018 Annual American Control Conference (ACC)* (2018).
- [2] G. Belgioioso D. Liao-McPherson M.H. de Badyn S. Bolognani J. Lygeros and F. Dörfler. “Sampled-data online feedback equilibrium seeking: stability and tracking”. In: (2021).
- [3] M. Bianchi G. Belgioioso and S. Grammatico. “Fast generalized Nash equilibrium seeking under partial-decision information”. In: *arXiv preprint arXiv:2003.09335* (2020).
- [4] V. Häberle A. Hauswirth L. Ortmann S. Bolognani and F. Dörfler. “Non-Convex Feedback Optimization with Input and Output Constraints”. In: *IEEE Control Systems Letters* 5.1 (2020), pp. 343–348.
- [5] A. Bernstein E. Dall’Anese and A. Simonetto. “Online Primal-Dual Methods with Measurement Feedback for Time-Varying Convex Optimization”. In: *IEEE Transactions on Signal Processing* 67.8 (2019), pp. 1978–1991.
- [6] M. Colombino E. Dall’Anese and A. Bernstein. “Online Optimization as a Feedback Controller: Stability and Tracking”. In: *IEEE Transaction on Control of Network Systems* 7.1 (2019), pp. 422–432.
- [7] A.H. De Ruiter C. Damaren and J.R. Forbes. *Spacecraft Dynamics and Control: An Introduction*. John Wiley & Sons, 2012.
- [8] A. Hauswirth S. Bolognani G. Hug F. Dörfler. “Optimization Algorithms as Robust Feedback Controllers”. In: (2021). preprint.
- [9] A.Y. Yazicioglu M. Egerstedt and J.S. Shamma. “Communication-Free Distributed Coverage for Networked Systems”. In: *IEEE Transactions on Control of Network System* 4.3 (2017).
- [10] D. Gadjov and L. Pavel. “A Passivity-based Approach to Nash Equilibrium Seeking over Networks”. In: *IEEE Transaction on Automatic Control* 64.3 (2018), pp. 1077–1092.
- [11] K. Alfriend S.R. Vadali P. Gurfil J. How and L. Breger. *Spacecraft Formation Flying: Dynamics, Control and Navigation*. Vol. 2. Butterworth-Heinemann, 2009.
- [12] H. Jaleel and J.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.
- [13] M.S. Stankovic K.H. Johansson and D.M. Stipanovic. “Distributed Seeking of Nash Equilibria with Applications to Mobile Sensor Networks”. In: *IEEE Transaction on Automatic Control* 57.4 (2011), pp. 904–919.
- [14] S.M. LaValle and S.A. Hutchinson. “Game Theory as a Unifying Structure for a Variety of Robot Tasks”. In: *Proc. 8th IEEE Int. Symp. Intell. Control* (1993), pp. 429–434.

- [15] A. Li and J. Mason. “Optimal Utility of Satellite Constellation Separation with Differential Drag”. In: *AIAA/AAS Astrodynamics Specialist Conference* (2014).
- [16] C.R. McInnes. “Autonomous Ring Formation for a Planar Constellation of Satellites”. In: *Journal of Guidance, Control and Dynamics* 18.5 (1995), pp. 1215–1217.
- [17] R.M. Murray. “Lecture 2 - LQR Control”. In: () .
- [18] A.R. Romano and L. Pavel. “Dynamic NE Seeking for Multi-Integrator Networked Agents with Disturbance Rejection”. In: *IEEE Transaction on Control of Networked Systems* 7.1 (2019), pp. 129–139.
- [19] A.M.K. Hanafi E. Hachem T. Rachidi M. Sahmoudi. “Perturbation Effects in Orbital Elements of CubeSats”. In: *3rd International Conference on Advanced Technologies for Signal and Image Processing* (2017).
- [20] Y. Ulybyshev. “Long-term formation keeping of satellite constellation using linear-quadratic controller”. In: *Journal of Guidance, Control, and Dynamics* 21.1 (1998), pp. 109–115.
- [21] R. Zhang F. Wang and L. Guo. “On Game-based Control Systems and Beyond”. In: *National Science Review* 7 (2020), pp. 1116–1117.
- [22] J. Williams. “LVLH Transformations”. In: *IEEE Spectrum* (2014).
- [23] P. Yi and L. Pavel. “An Operator Splitting Approach for Distributed Generalized Nash Equilibria Computation”. In: *Automatica* 102 (2019), pp. 111–121.
- [24] E. Sin H. Yin and M. Arcak. “Passivity-based distributed acquisition and station-keeping control of a satellite constellation in aerostationary orbit”. In: *arXiv preprint arXiv:2005.12214* (2020).