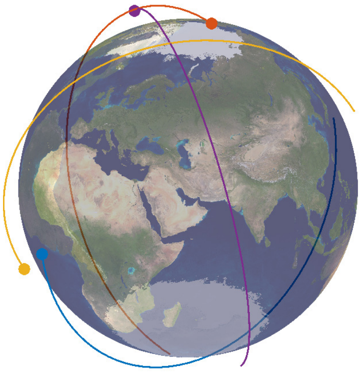


Collaborative Localization for Satellite Formations

João Francisco Marafuz Gaspar

PIC2 report to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. João Pedro Castilho Pereira Santos Gomes
Prof. João Manuel de Freitas Xavier

January 2024

Contents

Chapter 1

Introduction

The Earth is the cradle of humanity, but mankind cannot stay in the cradle forever.

Konstantin E. Tsiolkovsky

This first chapter presents the motivation for studying satellite systems and navigation, followed by an overview of the current state-of-the-art. It then highlights the objectives, deliverables, and work plan, concluding with a report outline that previews the topics covered.

1.1 Motivation

Over the last few decades, multi-agent systems, *i.e.*, networks of multiple singular nodes with local data gathering, processing, and communication capabilities [1], have attracted considerable interest, particularly in the realm of space exploration, namely, the concept of spacecraft Formation Flying (FF). Also, a great focus has been given to the goal of reducing the cost of space exploration missions, for which, among all the new concepts and developments being studied for this purpose, FF may prove to be one of the most important technological shifts. FF enables the division of payload and distribution of processing tasks among multiple smaller and more cost-effective units, effectively lowering the size and cost of the devices [2]. By transitioning from a single, large monolithic satellite to an assembly of smaller satellites, FF not only aims to cut costs but also enhances mission reliability, as the malfunction or loss of a single satellite does not necessarily lead to the collapse of the system [3]. Moreover, FF broadens the scope of potential scientific missions, enabling endeavors that would be impractical with just one spacecraft [4]. However, a key challenge in FF lies in designing a robust navigation system [5], which is critically dependent on accurate Orbit Determination (OD) techniques.

OD is essential for accurately estimating the orbits of space objects, such as Earth-orbiting satellites, in terms of position, velocity, or other equivalent state spaces, based on the available measurements [6, 7]. The history of OD is deeply rooted in the field of astronomy, particularly in the prediction of planetary and cometary motions. Notable early contributors to this field include Copernicus (1473 – 1543), Kepler (1571 – 1630), Newton (1642 – 1727), Lagrange (1736 – 1813), and Gauss (1777 – 1855), whose pioneering work laid the foundation for the techniques and methods used in modern OD [8]. The technological advancements in recent decades, particularly in computational power and sensor technology, have significantly enhanced OD's precision. This evolution is exemplified by the tracking of Sputnik in 1957, initially achieved with visual observations limited to a few kilometers in accuracy, progressing to contemporary OD solutions achieving sub-centimeter accuracy [7]. Furthermore, the efficacy of space missions often hinges on a satellite's ability to adhere to a specified nominal orbit, necessitating corrective maneuvers when deviations occur, thereby underlining the crucial role of OD in orbit control strategies.

As the number of satellites in an FF grows, the classical OD estimation solutions, which are developed under a centralized framework, eventually become infeasible to implement in practice – a phenomenon often referred to as the curse of dimensionality. These traditional centralized frameworks require: i) an infrastructure dedicated to centralized coordination; ii) transmission of substantial amounts of data between each satellite and a central node; and iii) significant computational power for real-time processing at this central node. With the expansion of the network, this approach faces increasing challenges, including heavier loads on communication links, longer communication delays, more complex protocols, and greater demands on the central unit's computational capacity. These issues make centralized implementation impractical in larger FFs and also expose vulnerabilities, such as reduced robustness to failures in the central processing node or communication systems. [9]

Recognizing these limitations, there has been a shift towards decentralized and distributed estimation frameworks for OD, to mitigate the curse of dimensionality. In such setups, no central computing unit is required, and no single entity within the FF possesses complete knowledge of the network's global state. Decentralized estimation frameworks emphasize local computations and communications while utilizing the same physical data transmission infrastructure as a centralized framework where the key distinction lies in the data management at the protocol level. Here, only locally pertinent information is transmitted, preventing the data overload that plagues centralized frameworks as the FF size increases. On the other hand, distributed estimation frameworks involve a higher degree of cooperation and information sharing among satellites. Each satellite in a distributed network contributes to the overall processing task, collaboratively working towards a common goal. This collaborative nature often results in more robust and efficient processing, as data and computational responsibilities are shared across the network, leading to optimized resource utilization and potentially higher resilience against failures on individual satellites. For instance, if an estimation solution employs local estimation methods in each satellite, but the parameters for these estimation algorithms have to be computed in real time by a single satellite or replicated across every satellite in the constellation, the estimation solution is decentralized but not distributed. Thus, the inevitable paradigm change towards large-scale FF calls for distributed and decentralized OD algorithms.

1.2 Literature Review

Research into the development and testing of spacecraft Guidance, Navigation, and Control (GNC) systems has been highly active since its inception in the early 1960s, so this section is dedicated to showcasing significant literature and studies that focus on navigation systems, and OD and FF techniques. The state-of-the-art technologies established within this section form the basis upon which novel navigation algorithms can be proposed. Thus, before diving into the objectives for this work, it is essential to first examine the historical and contemporary strategies in this field, along with their inherent challenges.

In recent times, there has been a surge in literature studies of spacecraft FF. Notably, a survey by

Di Mauro *et al.* [5] presents an overview that includes 233 FF missions scheduled from 2000 to 2025. In a similar vein, a survey by Bandyopadhyay *et al.* [10] categorize several missions involving small satellites, specifically those with a wet mass under $10\,\mathrm{kg}$. These comprehensive reviews serve as a solid foundation for understanding the evolution and variety of FF missions. The remainder of this section addresses particular aspects of the spacecraft navigation problem and OD algorithms for spacecraft FF.

The Evolution of Orbit Determination Techniques

Historically, spacecraft position and velocity determination primarily relied on methods focusing on individual spacecraft. These methods aimed to estimate the spacecraft's state concerning Earth referred to as the absolute spacecraft state. These OD processes traditionally utilized statistical or regression-based methods for post-processing on-ground observations over extended periods. This methodology, often known as batch processing [11], typically employs algorithms like Least-Squares and its variants. These algorithms deduce the spacecraft's orbit parameters based on a series of measurements. The spacecraft's known position is then propagated forward in time using a dynamics model until sufficient data is acquired for the next batch estimation. However, this approach has limitations: the extended duration required for data collection leads to incremental errors from the dynamics model during state propagation, reducing confidence in real-time state estimates until the subsequent batch is processed. Additionally, batch processing demands substantial computational resources at the terrestrial central node. Despite its limited real-time state estimates, batch processing's accuracy remains high, as exemplified by the GRACE mission (2002) [5]. In this early two-satellite FF example, neither spacecraft had onboard navigation systems, and all navigation computations were ground-based, using GPS measurements. The mission did not require autonomous navigation due to the separation distance between the spacecraft $(170 - 270\,\mathrm{km})$.

Conversely, on-board processing in FF at least necessitates a "leader" spacecraft, sometimes referred to as the chief, equipped with more advanced capabilities, such as a Global Navigation Satellite System (GNSS) receiver for absolute state estimation. The remaining FF spacecraft, termed followers or deputies, possess less expensive sensing capabilities and depend on relative range measurements to the leader for position estimation. This configuration is evident in forthcoming missions like NASA's Helioswarm nine-spacecraft mission (2028) [12]. These missions present complex navigation or localization challenges due to the scale of the estimation problem – the curse of dimensionality. Consequently, implementing an agent-specific on-board localization system that distributes the workload becomes essential. This approach balances computational costs among all FF agents and manages inter-spacecraft communications. Moreover, an on-board OD algorithm providing real-time data with lower computational demands is necessary. The TanDEM-X and TerraSAR-X missions (2010) [5] exemplify this, utilizing real-time state estimates and incorporating sequential estimation methods in their navigation systems. This method provided immediate data for guidance and control, with more accurate batch processing performed later on the ground to refine the baseline separation distance critical for their scientific instruments. Among various sequential estimation techniques, the Extended Kalman Filter (EKF) is the predominant choice for real-time navigation in spacecraft.

Kalman Filter-based Orbit Determination Techniques

The theoretical details of the Kalman Filter (KF) will be presented in a subsequent chapter of this work, so a review of several high-profile FF missions that used EKF-based navigation systems is given here to demonstrate the flight heritage and on-orbit performance capabilities of the EKF. Firstly, one of the most promising demonstrations of autonomous FF technologies to date has been the dual-satellite PRISMA mission (2010) [5]. PRISMA consists of a leader/follower coordination approach for which on-orbit experiments demonstrated relative navigation accuracy on the centimeter level for minimum separations of $150\,\mathrm{m}$ using an EKF [5, 13]. An experiment entitled Formation Flying In-Orbit Ranging Demonstration (FFIORD) [5] utilized the PRISMA spacecraft to validate an FF radio frequency sensor package. An EKF was used during the FFIORD testing to provide estimates of the relative spacecraft positions and velocities; furthermore, several onboard dynamics models were tested within the EKF. The CanX mission [5] launched a dual spacecraft formation in 2014, consisting of the Can-X4 and Can-X5 spacecraft, respectively the chief and the deputy. The deputy uses a relative navigation algorithm based on an EKF that uses carrier phase differential GPS techniques to estimate the relative state of the deputy concerning the chief. The MMS mission (2015) [5] developed by NASA launched four spacecraft into a tetrahedral formation. Precise formation-keeping control was required during science-collection operations, which required knowledge of the spacecraft separation to within $100\,\mathrm{m}$. This accuracy was accomplished by onboard navigation software that used Global Positioning System (GPS) measurements, an EKF, and a high-fidelity onboard dynamics model.

Although the EKF has proven to be quite effective in spacecraft applications, there are still limitations that can compromise the accuracy of the final state estimates. Deficiencies in the mathematical model used internally by the filter to describe the real-world system can lead to filter divergence over time if the modeling mismatch is too large and will further degrade the performance of the EKF if left unaddressed. A subset of Kalman filtering research has thus investigated ways to modify the performance of the standard EKF in real-time based on observations of the filter performance. Fraser [14] proposed two novel adaptive Kalman filtering techniques: one uses maximum likelihood estimation techniques to derive analytical adaptation laws for the filter, which are then improved through the novel inclusion of an intrinsic smoothing routine and the second uses an embedded fuzzy logic system based on a covariance-matching analysis of the filter residuals, where the fuzzy system has been specifically designed for the spacecraft navigation problem at hand. These demonstrated to be appreciably more robust to filter initialization errors, dynamics modeling deficiencies, and measurement noise than the centralized EKF. Also, Oliveira [15] introduces the Extended Semi-analytical Kalman Filter and a novel Unscented Semi-analytical Kalman Filter, integrating semi-analytical propagation with Kalman filtering techniques.

Another challenge arises with the use of EKF for OD in spacecraft FF — as the state size of the network to be estimated increases, the computational demands on the leader spacecraft escalate significantly. To address this, research has shifted towards reducing this computational burden by leveraging the communication systems among spacecraft in the FF, leading to the development of distributed localization algorithms.

Decentralized and Distributed Orbit Determination Techniques

In the realm of spacecraft orbit determination and navigation, decentralized and distributed estimation strategies are increasingly gaining prominence. A significant breakthrough in this field is the consensus EKF for distributed state estimation. This approach, as detailed in references [16] and [17], utilizes the concept of consensus in networked nonlinear dynamical systems to fuse prior and novel information across nodes. This methodology has shown promising results in enhancing system observability and mitigating the convergence on ambiguous orbits, thereby improving the overall performance of relative motion estimation in spacecraft. Another pivotal contribution to this field is the Distributed Kalman Filter, as outlined in references [18] and [19]. These papers discuss the decomposition of a central Kalman filter into micro-Kalman filters across a network, where consensus filters play a crucial role in facilitating the average consensus of time-varying signals. This approach not only simplifies the computational process but for systems described by linear state-space models it also maintains the estimation accuracy akin to that of a central Kalman filter. These algorithms demonstrate comparable efficiency and accuracy with established algorithms, highlighting their potential in satellite ephemeris[1] determination.

A novel perspective on decentralized parameter estimation is provided in [20]. The proposed algorithm merges local stochastic approximation methods with a global consensus strategy, effectively reducing measurement noise and enhancing estimation accuracy. This is particularly beneficial in multi-agent systems where communication noise can significantly impact the system's performance. Reference [21] introduces an integrated approach to GNC of spacecraft FF. This approach amalgamates a decentralized filter with Covariance Intersection, thereby efficiently estimating the full relative state of spacecraft formations while avoiding the divergence issues commonly associated with EKF. The scalability of localization algorithms in spacecraft swarms is addressed in [22]. The Decentralized Pose Estimation algorithm presented here takes into account both communication and relative sensing networks, allowing each spacecraft to localize a subset of the formation. This approach ensures that the complexity of the algorithm does not increase with the number of spacecraft, making it highly scalable and efficient for large swarms.

In the Bayesian framework of estimation, the work presented in [23] introduces a distributed state estimator based on the consensus on Kullback-Leibler averages of local probability density functions. This method guarantees mean-square boundedness of the state estimation error, ensuring stability under minimal network connectivity and system observability requirements. References [24, 25] explore consensus-based algorithms for distributed filtering, combining Consensus on Information (CI) and Measurements (CM) to form Hybrid (HCMCI) consensus filters. These filters capitalize on the strengths of both CI and CM, providing a stable and effective solution for distributed state estimation in linear systems. The hybrid approach is validated through simulation case studies, demonstrating its applicability in various sensor network scenarios. Lastly, [26] discusses the Kalman-Consensus Filter (KCF), highlighting its optimal decentralized structure and scalability in sensor networks. The KCF algorithm efficiently estimates the state of dynamic targets and reaches a consensus with neighboring estimator agents,

[1] The set of data broadcast by a satellite that provides detailed information about its position and trajectory in orbit relative to the Earth, including its location, velocity, and timing information, essential for accurate navigation and positioning systems.

exemplifying a significant advancement in sensor networks with variable topology.

To date, however, there have been no widely accepted or reported distributed techniques implemented on spacecraft in orbit, and the development and demonstration of distributed methods have therefore been restricted to simulation environments, hardware-in-the-loop testing, and theoretical propositions.

Other Orbit Determination Techniques

Recent advancements in orbit determination techniques have emphasized the need for efficient communication and computational strategies, particularly in small-satellite formations. Cordeiro's work [27] introduces a distinctive distributed state estimation framework for such formations. It addresses the limitations in inter-satellite communication by adopting a leader-follower topology. Here, the leader satellite accesses absolute position measurements, while the followers rely on relative range measurements. The Consider Covariance (CC) technique is used to include the leader state uncertainty in the follower's filter; more on this technique later in this work. This framework, notably distinct for its decoupling of the leader's state estimation from the followers', showcases reduced communication requirements and enhanced robustness, particularly in high-leader uncertainty scenarios. This technique will be flight-tested on an upcoming NASA small-satellite mission consisting of four coffee mug-sized satellites based on open-source PyCubed flight hardware, each containing high-accuracy GPS and time-of-flight ranging.

In the domain of distributed estimation algorithms, Lopes, Cattivelli and Sayed [28–30] have made significant contributions. Lopes [28] delves into diffusion protocols in adaptive networks, demonstrating their superiority in terms of mean-square error over non-cooperative schemes. Cattivelli [29] presents diffusion LMS strategies for distributed estimation, introducing algorithms that show improved performance and robustness. Further extending this work, Cattivelli [30] focuses on distributed Kalman filtering and smoothing, proposing algorithms for collaborative estimation that leverage the strengths of diffusion strategies.

Lastly, Jia's research [31] introduces the DeCiSpOT filter, a novel approach to cooperative space object tracking. This filter balances computational complexity and communication needs between sensors, effectively enhancing track accuracy in challenging measurement scenarios. The DeCiSpOT algorithm employs innovative strategies like the global nearest neighbor and probabilistic data association, proving its effectiveness in simulated space object tracking scenarios.

1.3 Objectives, Deliverables, and Work Plan

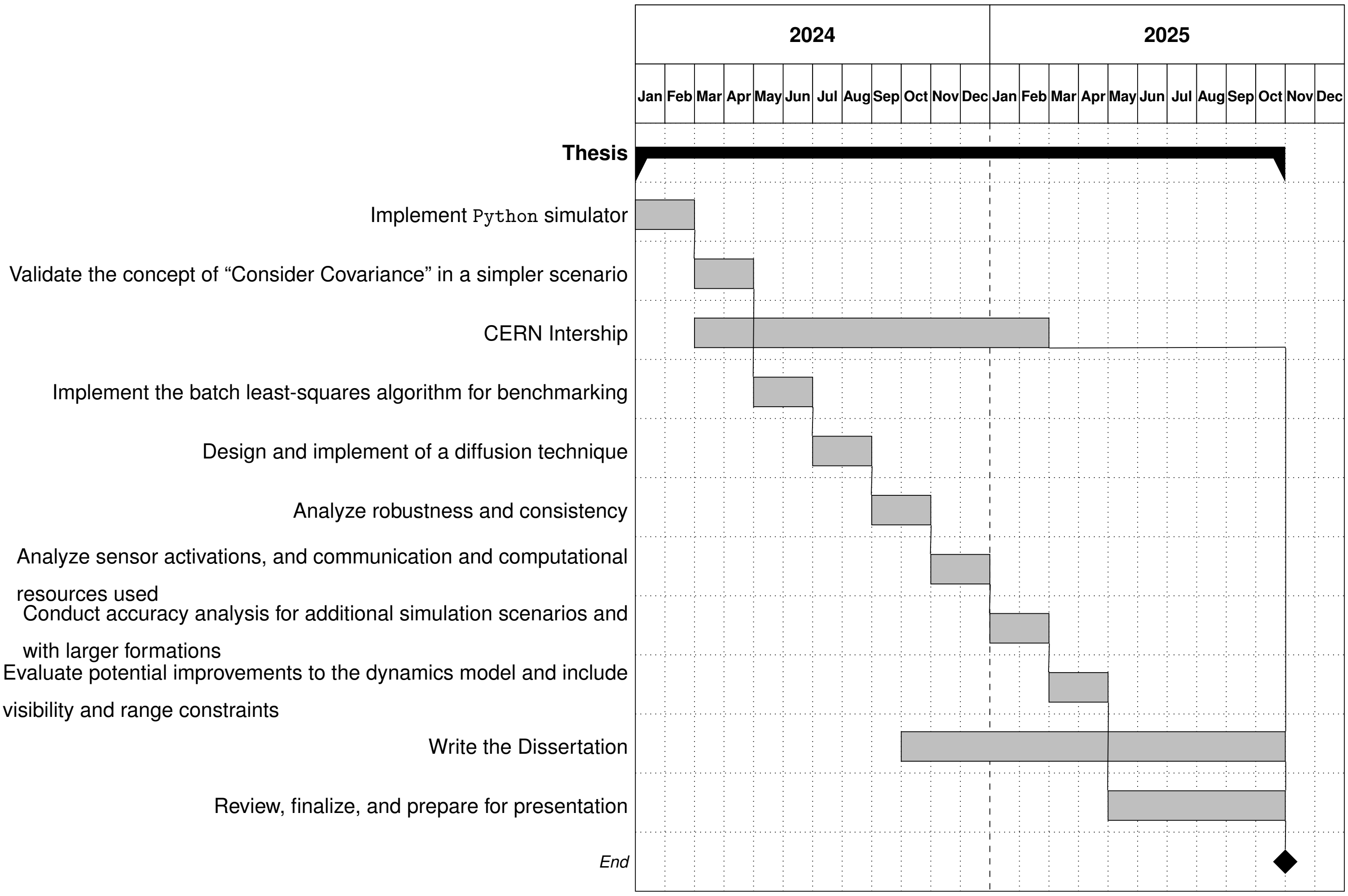
The main objective of this thesis is to implement a diffusion technique on a scalable flying formation as a way to improve the accuracy of the orbit determination procedures for big constellations, after exploring the pre-existing ones. To achieve this, a list of objectives was defined:

- Develop a 3D satellite dynamics simulator in `Python` (here), capable of simulating the most relevant orbital perturbations. This simulator should be able to handle multiple satellites simultaneously and include an animation feature using a `MATLAB` graphical interface.

- Develop and implement both a centralized and a consensus-based solution for the orbit determination problem in Python, to be used as benchmarks.
- Interpret, implement, and validate the concept of Consider Covariance, as introduced in the solution for this problem in [27]. Use this also as a benchmark.
- Design an algorithm that is not redundant in terms of computational and communication resources, scalable, avoids the inner loop of consensus, and primarily leverages the distribution and exchange of information among formation flying elements through communication.

To achieve such goals a Gantt chart with the proposed work split is presented in Table 1.1.

Table 1.1: Gantt Chart of the Work Plan.



1.4 Report Outline

This document is organized as follows. Chapter 2 provides background information, including spacecraft formation flying, orbital mechanics covering the theoretical foundations for this work, and estimation algorithms for orbit determination. Chapter 3 discusses the implementation details, such as the problem statement and the modeling of dynamics and observations, along with the formulation of a fully centralized solution, a consensus-based solution, and a consider covariance-based one. Finally, preliminary results for the implementation of the previously described orbit determination algorithms are presented in Chapter 4.

Chapter 2

Background

That's one small step for man, one giant leap for mankind.

Neil Armstrong

This chapter focuses on the theoretical principles that support the concepts explored in the dissertation. Firstly, several orbital mechanics ground concepts are introduced and discussed. Following this, the studied estimation techniques are presented, as well as the specific case of consider covariance. Finally, the chapter delves into diffusion approaches and how they can be integrated into orbit determination and parameter estimation.

2.1 Spacecraft Formation Flying

The concept of spacecraft FF lacks a precise or universally accepted definition. However, a widely recognized description, as proposed by NASA's Goddard Space Flight Center and mentioned in [2], describes FF as "The tracking or maintenance of a desired relative separation, orientation or position between or among spacecraft".

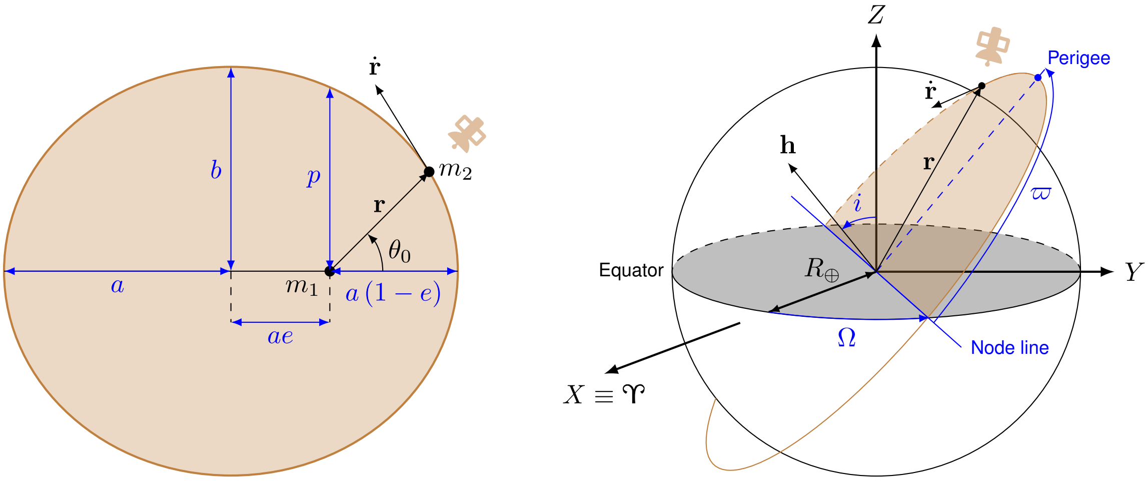
Since the 1990s, numerous missions have utilized FF in fields like astronomy, communications, and weather observation. These missions have not only showcased the practicality of FF but also corroborated its advantages and significance when compared with the existing advantages. Key advantages of this approach are outlined in a survey by Liu (2018) [3]:

- **System Cost Reduction:** Using small, networked satellites for space missions significantly lowers costs. Standardized production processes make these smaller spacecraft cheaper to design and manufacture than larger ones. Their reduced size and weight also decrease launch expenses. Furthermore, if a satellite malfunctions, its replacement is both cost-effective and swift, further reducing overall system maintenance costs.
- **System Performance Improvement:** The use of multiple satellites in a networked formation enhances system robustness and efficiency through resource redundancy. This setup promotes autonomous navigation and control, reduces reliance on ground stations, and improves the overall autonomy and intelligence of the system. Additionally, distributing and parallelizing tasks across the formation boosts efficiency.
- **System Reliability Enhancement:** The modular design of a networked satellite formation, featuring standardized communication and control components, ensures system stability. This design allows for continued operation even if individual satellites fail, as only the links to those specific satellites are affected, preventing a complete system collapse.

2.2 Orbital Mechanics

As a fundamental aspect of this work, it is imperative to introduce and discuss key concepts of orbital mechanics. This will provide the reader with essential background information on the dynamic interaction between a central body and an orbiting satellite. In this context, the central body, namely planet Earth, and the satellite are denoted as masses $m_1 = M_\oplus$[1] and m_2, respectively, as illustrated in Figure 2.1.

Initially, it's essential to define an inertial reference frame. Due to their proximity, satellites orbiting Earth are subject to similar solar acceleration as the Earth itself. This similarity allows for the neglection of solar acceleration effects on both the Earth and the satellite, rendering the system effectively inertial in terms of translation. The reference frame used to describe this condition is termed the Earth-Centered Inertial (ECI) frame [32].



(a) 2D representation.

(b) 3D representation.

Figure 2.1: 2D and 3D representations of a satellite orbit around Earth and its classical orbital elements.

The subsequent step involves determining the rotation of the reference frame and establishing the orientation of its coordinate axes. Given the assumption that the Universe does not rotate, the orientation of the reference frame's axes can be fixed based on observations of distant stars. Accordingly, the Z axis aligns with Earth's rotational axis, extending from south to north. The X axis is oriented towards the Vernal Equinox, marking the point where the ecliptic intersects the equator and the Sun transitions from the Southern to the Northern Hemisphere. The Y axis, orthogonal to both X and Z, forms a right-handed coordinate system and lies in the equatorial plane. Historically, the X axis was aligned with the constellation Aries, known as the First Point of Aries, and is denoted as Υ. This reference frame will be used throughout this work. [8, 32–35]

2.2.1 Classical Orbital Elements

To specify an orbit in space, two key vectors are essential: the satellite's position vector, denoted as $\mathbf{r}$, and its velocity vector, denoted as $\dot{\mathbf{r}}$. These vectors must be known at a specific moment in

[1] The symbol $\oplus$ is commonly used to refer to Earth.

time. However, specifying the set $\{\mathbf{r}, \dot{\mathbf{r}}\}$, which comprises 6 parameters, is not practically efficient for delineating the satellite's orbit. Instead, within an ECI frame, it is more convenient to describe the satellite's orbit using a different set of 6 parameters. These are known as the classical orbital elements. They effectively determine the satellite's trajectory and its orientation in space. [32]

They are divided into two subsets, those that specify the orbit's orientation in space (see Figure 2.1b):

- The **Inclination** i of the orbit, is the angle between the orbit plane's normal $\mathbf{h}$ and the polar axis Z. It is limited to the interval $i \in [0, \pi]$ rad to ensure a bi-univocal correspondence.
- The **Right Ascension of the Ascending Node** Ω, the angle between the X-axis and the intersection of the orbit with the equator, on the side where the satellite passes from the Southern Hemisphere to the Northern Hemisphere. It is limited to the set $\Omega \in [0, 2\pi]$ rad.
- The **Argument of Perigee** ϖ, the angle measured from the ascending node, in the orbital plane, until reaching the perigee, which is the nearest orbital point from the Earth on the semi-major axis. As Ω it is also limited to the set $\varpi \in [0, 2\pi]$ rad.

The remaining three elements of the second subset relate to the type of orbit and the satellite's location in it, (see Figure 2.1a):

- The orbit is completely determined by the **Semi-major Axis** a and the **Eccentricity** e.
- The satellite's position in the orbit can be calculated at any instant by knowing the **Time of Passage at Perigee** T_0. It serves to relate the orbit to the passage of calendar time and allows for the calculation of the position in the orbit.

The classical orbital elements $\{i, \Omega, \varpi, a, e, T_0\}$ are much easier and more naturally interpreted than specifying $\{\mathbf{r}, \dot{\mathbf{r}}\}$ for a given time. Note that the true anomaly θ_0 (see Subfigure 2.1a) will be used instead of T_0, since it directly relates to the spacecraft's position in its orbit.

2.2.2 The Two-Body Problem

The most important contribution influencing the equations that govern the evolution of a satellite's velocity over time is encapsulated by the central force term of the Earth's gravitational attraction, represented by

$$\ddot{\mathbf{r}}_{\text{grav}} = -\frac{\mu_\oplus}{r^3}\mathbf{r}, \tag{2.1}$$

where $r = ||\mathbf{r}||_2$ represents the magnitude of $\mathbf{r}$, and $||\cdot||_2$ denotes the ℓ_2-norm. (2.1) is famously known as the two-body equation of relative motion whose solution is a conic section [32], in which $\mu_\oplus = 3.986 \times 10^5\,\text{km}^3\,\text{s}^{-2}$ [36] is known as Earth's gravitational parameter.

Remark that for (2.1) to be valid, the following assumptions must be met [14]: **Assumption 1 – The Massive Primary)** The mass of the orbiting satellite is much smaller and therefore negligible when compared to the mass of the central body; **Assumption 2 – Newtonian Dynamics)** The motion of the two bodies is described exclusively by the mutually attractive Newtonian gravitational forces; **Assumption**

3 – Spherical Bodies) Both bodies are treated as particle point masses, following the assumption that they are spherically symmetric. However, several approximations need to be made to consider these assumptions. This will be approached in Subsection 2.2.3.

Now, using the general relation

$$\frac{\partial r^n}{\partial \mathbf{r}} = \frac{\partial \left(x^2 + y^2 + z^2\right)^{n/2}}{\partial \mathbf{r}} = n r^{n-2} \mathbf{r}^T, \tag{2.2}$$

which yields a row vector, it follows that the partial derivative

$$\frac{\partial \ddot{\mathbf{r}}_{\text{grav}}}{\partial \mathbf{r}} = -\mu_\oplus \frac{\partial}{\partial \mathbf{r}} \left(\mathbf{r} \frac{1}{r^3}\right) = -\mu_\oplus \left(\frac{1}{r^3} \mathbf{I}_3 - 3 \frac{\mathbf{r}\mathbf{r}^T}{r^5}\right). \tag{2.3}$$

Here the factor $\mathbf{r}\mathbf{r}^T$ in the second term is an outer product, which yields a matrix and should not be confused with the dot product $\mathbf{r}^T\mathbf{r}$.

2.2.3 Orbital Perturbations

The two-body equation (2.1) offers a basic understanding of spacecraft motion but relies on assumptions not true in reality. Earth isn't a perfect sphere but has the form of an oblate spheroid with an equatorial diameter that exceeds the polar diameter by about $20\,\mathrm{km}$ [35], and spacecraft often have complex shapes due to components like solar panels. Consequently, real spacecraft motion doesn't strictly follow Keplerian models due to various external forces like atmospheric drag (in case of sufficiently low orbits), solar radiation, and gravitational influences from the Earth, Moon, Sun and other bodies. This leads to perturbed or non-Keplerian motion. Table 2.1 represents the most important orbital perturbations for a Low Earth Orbit (LEO) satellite.

Table 2.1: Orbital perturbations' orders of magnitude for a LEO spacecraft ($r \in [160, 2000]\ \mathrm{km}$) with a perpendicular area to the acceleration $A_\perp$ and mass m. [35]

		Zonal harmonics				**Third-body**	
Orbital Perturbation	**2-body**	J_2	J_3	**Drag**	**Solar radiation**	**Moon**	**Sun**
	(max)	(max)	(max)	(max)	(max)	(mean)	(mean)
Acceleration $[\mathrm{km\,s^{-2}}]$	10^{-3} (2.1)	10^{-5} (2.6)	10^{-8} [37]	$10^{-7}\frac{A_\perp}{m}$	$10^{-9}\frac{A_\perp}{m}$	10^{-9}	10^{-10}

The two-body acceleration not surprisingly dominates the system, with the effects of the Earth's oblateness J_2 term contributing the largest perturbation, as described below in more detail. Similarly, the influence of drag contributes noticeably to the acceleration of a spacecraft. Since this research focuses on formations In LEO conditions, it is pertinent to incorporate both into the model.

Earth's Oblateness

The most significant perturbation in this scenario arises from Earth's non-spherical shape, which is represented by modeling the gravitational potential with an infinite series of Legendre polynomials. Though the most comprehensive expression is beyond the scope of this work, a simplified version will be provided. This version considers only the zonal harmonics, which account for variations in latitude and are valid when the mass distribution still has axial symmetry. This is sufficient to demonstrate the primary effect, which is the flattening of the poles. Thus, if the mass distribution has axial symmetry, the gravitational potential can be described by the series [8, 38]:

$$U(r,\phi) = \frac{\mu_\oplus}{r} - \frac{\mu_\oplus}{r}\sum_{k=2}^{\infty} J_k \left(\frac{R_\oplus}{r}\right)^k P_k\left(\sin\phi\right), \tag{2.4}$$

where ϕ represents the latitude with respect to the Equator, J_k contains the zonal harmonics of the planet, and $P_k(\cdot)$ are the Legendre polynomials. In (2.4), the first term is equivalent to the distribution with spherical symmetry and the second

$$U_{J_2}(r,\phi) = -\frac{\mu_\oplus}{r} J_2 \left(\frac{R_\oplus}{r}\right)^2 P_2\left(\sin\phi\right) = -\frac{3J_2\mu_\oplus}{2r}\left(\frac{R_\oplus}{r}\right)^2\left(\sin^2\phi - \frac{1}{3}\right) = -\frac{3J_2\mu_\oplus R_\oplus^2}{2r^3}\left(\frac{z}{r}\right)^2 + \frac{J_2\mu_\oplus R_\oplus^2}{2r^3} \tag{2.5}$$

describes the flattening of the body, since $J_2 = 0.00108262545$ [36] is over a thousand times larger than the next largest term. Remind that U_{J_2} was rewritten as a function of (r, z), since $\sin\phi = z/r$. Now, evaluating the gradient of (2.5), we can write the perturbing acceleration due to the oblateness of the Earth as [8]:

$$\ddot{\mathbf{r}}_{J_2} = -\frac{3J_2\mu_\oplus R_\oplus^2}{2r^5}\begin{bmatrix}\left(1-5\frac{z^2}{r^2}\right)x \\ \left(1-5\frac{z^2}{r^2}\right)y \\ \left(3-5\frac{z^2}{r^2}\right)z\end{bmatrix}. \tag{2.6}$$

For later use, the partial derivatives of the acceleration due to the Earth's oblateness with respect to the position vector are given by (computed with SymPy and confirmed with [27]):

$$\frac{\partial\ddot{\mathbf{r}}_{J_2}}{\partial\mathbf{r}} = -\frac{3J_2\mu_\oplus R_\oplus^2}{2r^9}\times \\ \times\begin{bmatrix} -4x^4-3x^2y^2+27x^2z^2+y^4-3y^2z^2-4z^4 & -5x^3y-5xy^3+30xyz^2 & -15x^3z-15xy^2z+20xz^3 \\ -5x^3y-5xy^3+30xyz^2 & x^4-3x^2y^2-3x^2z^2-4y^4+27y^2z^2-4z^4 & -15x^2yz-15y^3z+20yz^3 \\ -15x^3z-15xy^2z+20xz^3 & -15x^2yz-15y^3z+20yz^3 & 3x^4+6x^2y^2-24x^2z^2+3y^4-24y^2z^2+8z^4 \end{bmatrix}. \tag{2.7}$$

Atmospheric Drag

Until now, the discussion has focused on gravitational perturbations affecting spacecraft, which are conservative forces, that is, they can alter a spacecraft's orbit shape or orientation but not its overall energy [39]. In LEO, however, non-conservative atmospheric drag forces reduce the orbit's energy and semi-major axis, potentially leading to de-orbiting if not considered in mission design, especially below altitudes of $1000\,\mathrm{km}$ [40].

For a body moving with some relative velocity $\dot{\mathbf{r}}_{\mathrm{rel}}$ concerning the atmosphere, the drag force is given by [35]:

$$\ddot{\mathbf{r}}_{\mathrm{drag}} = -\frac{1}{2}C_D\frac{A_\perp}{m}\rho_{\mathrm{atm}}\dot{r}_{\mathrm{rel}}\dot{\mathbf{r}}_{\mathrm{rel}}, \quad \text{with} \quad \dot{\mathbf{r}}_{\mathrm{rel}} = \dot{\mathbf{r}} - \boldsymbol{\omega}_\oplus\times\mathbf{r} \quad \text{and} \quad \boldsymbol{\omega}_\oplus = \begin{bmatrix}0 & 0 & \omega_\oplus\end{bmatrix}^T, \tag{2.8}$$

where a first-order approximation of the relative velocity of the spacecraft concerning the atmosphere can be made by assuming that the atmosphere co-rotates with the Earth at an angular velocity $\omega_{\oplus} = 7.292\,115\,146\,7 \times 10^{-5}\,\mathrm{rad\,s^{-1}}$ [36]. Atmospheric drag opposes a spacecraft's motion, its force proportional to the frontal area $A_{\perp}$ and scaled by a drag coefficient $C_D \approx 2.2$ (using a flat plate model [8]), considering factors like fluid composition and temperature. The spacecraft's mass m and design also influence drag effects and additionally, atmospheric density ρ_{atm} significantly affects drag forces.

To model ρ_{atm}, the exponential model described in [8] is used. This simple, static model assumes the density of the atmosphere decays exponentially with increasing altitude. It also assumes a spherically symmetrical distribution of particles, in which the density varies exponentially according to

$$\rho_{\text{atm}}(h) = \rho_0 \exp\left(-\frac{h - h_0}{H}\right), \tag{2.9}$$

where ρ_0 is the reference density, h_0 is the reference altitude, $h = r - R_{\oplus}$ is the actual altitude, above the ellipsoid, and H is the scale height[2]. This model, while approximating a significant portion of the atmosphere, lacks the precision required for highly accurate studies. To address this shortcoming, we have segregated the different altitude levels into closely related bands. This segregation enhances the model's accuracy, as the choice of bands plays a crucial role in determining the level of precision. The atmospheric parameters for these bands are detailed in Table 2.2.

Table 2.2: Exponential atmospheric model presented in [8].

h [km]	h_0 [km]	ρ_0 [kg m^{-3}]	H [km]	h [km]	h_0 [km]	ρ_0 [kg m^{-3}]	H [km]
$[0, 25)$	0	1.225	7.249	$[150, 180)$	150	2.070×10^{-9}	22.523
$[25, 30)$	25	3.899×10^{-2}	6.349	$[180, 200)$	180	5.464×10^{-10}	29.740
$[30, 40)$	30	1.774×10^{-2}	6.682	$[200, 250)$	200	2.789×10^{-10}	37.105
$[40, 50)$	40	3.972×10^{-3}	7.554	$[250, 300)$	250	7.248×10^{-11}	45.546
$[50, 60)$	50	1.057×10^{-3}	8.382	$[300, 350)$	300	2.418×10^{-11}	53.628
$[60, 70)$	60	3.206×10^{-4}	7.714	$[350, 400)$	350	9.518×10^{-12}	53.298
$[70, 80)$	70	8.770×10^{-5}	6.549	$[400, 450)$	400	3.725×10^{-12}	58.515
$[80, 90)$	80	1.905×10^{-5}	5.799	$[450, 500)$	450	1.585×10^{-12}	60.828
$[90, 100)$	90	3.396×10^{-6}	5.382	$[500, 600)$	500	6.967×10^{-13}	63.822
$[100, 110)$	100	5.297×10^{-7}	5.877	$[600, 700)$	600	1.454×10^{-13}	71.835
$[110, 120)$	110	9.661×10^{-8}	7.263	$[700, 800)$	700	3.614×10^{-14}	88.667
$[120, 130)$	120	2.438×10^{-8}	9.473	$[800, 900)$	800	1.170×10^{-14}	124.636
$[130, 140)$	130	8.484×10^{-9}	12.636	$[900, 1000)$	900	5.245×10^{-15}	181.065
$[140, 150)$	140	3.845×10^{-9}	16.149	$(1000, \infty)$	1000	3.019×10^{-15}	268.000

For later use, the partial derivatives of the acceleration due to atmospheric drag with respect to the velocity vector, using (2.2), are given by [35]:

$$\frac{\partial \ddot{\mathbf{r}}_{\text{drag}}}{\partial \dot{\mathbf{r}}} = -\frac{1}{2} C_D \frac{A_{\perp}}{m} \rho_{\text{atm}} \left(\frac{\dot{\mathbf{r}}_{\text{rel}} \dot{\mathbf{r}}_{\text{rel}}^T}{\dot{r}_{\text{rel}}} + \dot{r}_{\text{rel}} \mathbf{I}_3 \right). \tag{2.10}$$

The partial derivatives concerning the position involve a direct term describing the atmospheric density

[2] Scale height is the fractional change in density with height. It can be useful in determining numerical partial derivatives. [8]

variation as well as a minor contribution resulting from the changing atmospheric wind velocity [35]:

$$\frac{\partial \ddot{\mathbf{r}}_{\text{drag}}}{\partial \mathbf{r}} = -\frac{1}{2} C_D \frac{A_\perp}{m} \dot{r}_{\text{rel}} \dot{\mathbf{r}}_{\text{rel}} \frac{\partial \rho_{\text{atm}}}{\partial \mathbf{r}} - \frac{1}{2} C_D \frac{A_\perp}{m} \rho_{\text{atm}} \left(\frac{\dot{\mathbf{r}}_{\text{rel}} \dot{\mathbf{r}}_{\text{rel}}^T}{\dot{r}_{\text{rel}}} + \dot{r}_{\text{rel}} \mathbf{I}_3 \right) \frac{\partial \dot{\mathbf{r}}_{\text{rel}}}{\partial \mathbf{r}}, \tag{2.11}$$

where

$$\frac{\partial \rho_{\text{atm}}}{\partial \mathbf{r}} = -\frac{\rho_0}{H} \exp\left(-\frac{h - h_0}{H}\right) \frac{\partial r}{\partial \mathbf{r}} = -\frac{\rho_0}{H} \exp\left(-\frac{h - h_0}{H}\right) \frac{\mathbf{r}^T}{r} \tag{2.12}$$

and

$$\frac{\partial \dot{\mathbf{r}}_{\text{rel}}}{\partial \mathbf{r}} = -\frac{\partial}{\partial \mathbf{r}} \left(\boldsymbol{\omega}_\oplus \times \mathbf{r} \right) = -\frac{\partial}{\partial \mathbf{r}} \left(\begin{bmatrix} 0 & -\omega_\oplus & 0 \\ \omega_\oplus & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{r} \right) = \begin{bmatrix} 0 & \omega_\oplus & 0 \\ -\omega_\oplus & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.13}$$

2.3 Estimation Algorithms for Orbit Determination

This work will focus on the specific class of sequential methods for OD. These methods are forward-time recursions that repeat patterns of state prediction (orbit propagation) and measurement updates and are typically associated with real-time applications [6].

So, to effectively integrate measurements with a dynamic model for precise OD using relative sensing and non-linear dynamics, a filtering method like the EKF is essential. Moreover, to achieve a completely distributed system, it is necessary to enhance these estimates through mutual communication among spacecraft in the same formation. This can be accomplished by using consensus or diffusion strategies.

2.3.1 Extended Kalman Filter

The classic KF is the optimal recursive estimation algorithm that minimizes the expected value of the squared norm of the estimation error for linear dynamic system under Gaussian perturbations [41] and EKF corresponds to an extension (and approximation) of the KF for nonlinear systems, which is the case of satellite trajectories.

A nonlinear discrete-time system with no external actuation will be first considered in which the dynamics and observation models are respectively

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{w}, \tag{2.14}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}, \tag{2.15}$$

where $\mathbf{w}$ and $\mathbf{v}$ are their respective zero-mean white Gaussian noise components with covariance matrices $\mathrm{E}\left[\mathbf{w}\mathbf{w}^T\right] = \mathbf{Q}$ and $\mathrm{E}\left[\mathbf{v}\mathbf{v}^T\right] = \mathbf{R}$.

Given a set of system measurements $\boldsymbol{\mathcal{Y}}_{1:k} = \{\mathbf{y}_1, \ldots, \mathbf{y}_k\}$, the objective of the filter is to derive a refined estimate of the system's state by integrating these measurements with the dynamic propagation of the state vector. Unlike the standard KF, which computes the conditional mean of the Probability Density Function (PDF) of $\mathbf{x}_k$ given $\boldsymbol{\mathcal{Y}}_{1:k}$, in the presence of nonlinear dynamics, these conditional PDFs cease to be Gaussian. This alteration introduces significant computational challenges in propagating the entire PDF. The EKF addresses this issue by offering an approximation of the optimal estimate. It achieves this by linearizing the system's nonlinearities around the most recent state estimate. The EKF process involves linearization of the sequentially-predicted and filtered state estimates, as outlined in Algorithm 2.1.

Algorithm 2.1 Extended Kalman Filter. [41]

```
Require: Prior state $\hat{\mathbf{x}}_k$ and covariance $\mathbf{P}_k$ estimates obtained in the previous temporal step given observations up to, $\mathcal{Y}_{1:k}$, that will be corrected using the observation $\mathbf{y}_{k+1}$ of the current time step.
Ensure: Novel state $\hat{\mathbf{x}}_{k+1}$ and covariance $\mathbf{P}_{k+1}$ estimates.
Algorithm: EKF$(\hat{\mathbf{x}}_k, \mathbf{P}_k, \mathbf{y}_{k+1})$ return $\hat{\mathbf{x}}_{k+1}$, $\mathbf{P}_{k+1}$
    $\hat{\mathbf{x}}_{k+1|k} = \mathbf{f}\left(\hat{\mathbf{x}}_k\right), \quad \mathbf{F}_k = \frac{\mathrm{d}\mathbf{f}(\hat{\mathbf{x}}_k)}{\mathrm{d}\mathbf{x}}$ and $\mathbf{H}_{k+1} = \frac{\mathrm{d}\mathbf{h}\left(\hat{\mathbf{x}}_{k+1|k}\right)}{\mathrm{d}\mathbf{x}}$    ▷ Prediction:
    $\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}$
    $\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T \left(\mathbf{H}_{k+1} \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T + \mathbf{R}\right)^{-1}$    ▷ Kalman Gain
    $\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}\left[\mathbf{y}_{k+1} - \mathbf{h}\left(\hat{\mathbf{x}}_{k+1|k}\right)\right]$    ▷ Correction:
    $\mathbf{P}_{k+1} = \left(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1}\right)\mathbf{P}_{k+1|k}$
    return $\hat{\mathbf{x}}_{k+1}$, $\mathbf{P}_{k+1}$
```

In this algorithm, $\mathbf{P}_{k+1|k}$ and $\mathbf{P}_{k+1}$ represent the *a priori* and novel covariance matrices, respectively. Furthermore, $\mathbf{F}_k = \frac{\mathrm{d}\mathbf{f}(\hat{\mathbf{x}}_k)}{\mathrm{d}\mathbf{x}}$ and $\mathbf{H}_{k+1} = \frac{\mathrm{d}\mathbf{h}\left(\hat{\mathbf{x}}_{k+1|k}\right)}{\mathrm{d}\mathbf{x}}$ correspond to the Jacobians of the dynamics and measurement models, respectively. It is crucial to acknowledge that the EKF is not an optimal filter; it is based on a series of approximations. These approximations become particularly significant when the dynamic model is highly nonlinear, and the linearization around the previous estimate does not provide an accurate representation. Consequently, unlike the KF, the EKF might diverge if the successive linearizations do not adequately approximate the nonlinear model across all associated uncertainty domains.

2.3.2 Consider Covariance

Parameter errors in dynamic and measurement models of dynamic systems can result in poor state estimates when using a traditional KF structure. In dealing with these parameter errors it is possible to: 1) Ignore them completely; 2) Add the parameters as additional states to be estimated; or 3) "Consider" the error in the state covariance matrix by introducing additional parameter covariance matrices. To consider errors means to not estimate them directly, but instead update the state estimates and covariance based on an estimated error covariance. [42]

The consider covariance analysis technique helps evaluate the impact of omitting these unknown or potentially inaccurately modeled parameters on the state estimate's accuracy. This approach offers a more realistic assessment of the system's accuracy without the additional computational burden of including these parameters in the state model. In order to implement this, (2.14) and (2.15) need to be modified to include the consider parameters $\mathbf{c}_k$ as follows:

$$\mathbf{x}_{k+1} = \mathbf{f}_{\mathbf{x}}(\mathbf{x}_k) + \mathbf{f}_{\mathbf{c}}(\mathbf{c}_k) + \mathbf{w}, \tag{2.16}$$

$$\mathbf{y}_k = \mathbf{h}_{\mathbf{x}}(\mathbf{x}_k) + \mathbf{h}_{\mathbf{c}}(\mathbf{c}_k) + \mathbf{v}. \tag{2.17}$$

Consider the augmented state $\begin{bmatrix}\mathbf{x}^T & \mathbf{c}^T\end{bmatrix}^T$. Following this notation, the augmented covariance matrices, dynamic and measurement models Jacobians and augmented gain are

$$\mathcal{P} = \begin{bmatrix}\mathbf{P}_{\mathbf{xx}} & \mathbf{P}_{\mathbf{xc}} \\ \mathbf{P}_{\mathbf{cx}} & \mathbf{P}_{\mathbf{cc}}\end{bmatrix}, \quad \mathcal{Q} = \begin{bmatrix}\mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}, \quad \mathcal{R} = \mathbf{R}, \quad \mathcal{F} = \begin{bmatrix}\mathbf{F}_{\mathbf{x}} & \mathbf{F}_{\mathbf{c}} \\ \mathbf{0} & \mathbf{I}\end{bmatrix}, \quad \mathcal{H} = \begin{bmatrix}\mathbf{H}_{\mathbf{x}} & \mathbf{H}_{\mathbf{c}}\end{bmatrix}, \quad \mathcal{K} = \begin{bmatrix}\mathbf{K}_{\mathbf{x}} \\ \mathbf{0}\end{bmatrix}. \tag{2.18}$$

Since the consider parameters are not updated, the bottom right sub-Jacobian of $\mathcal{F}$ corresponds to the identity matrix and $\mathcal{K}$ is no longer the Kalman gain because some values were replaced with zeros. As a consequence, the general formula for the covariance update stage (see line 5 from Algorithm 2.1) needs to be remodeled with the Joseph form, which is a more general form of the covariance update equation. The Joseph form is given by [43]:

$$\mathcal{P}_{k+1} \triangleq \left(\mathbf{I} - \mathcal{K}_k \mathcal{H}_k\right) \mathcal{P}_{k+1|k} \left(\mathbf{I} - \mathcal{K}_k \mathcal{H}_k\right)^T + \mathcal{K}_k \mathcal{R} \mathcal{K}_k^T. \tag{2.19}$$

For implementation purposes, it is convenient to not create the augmented state, but instead maintain the consider parameter as an external variable and store both $\mathbf{P}_{\mathbf{xx}}$ and $\mathbf{P}_{\mathbf{xc}}$ separately with separate update equations. Therefore, the general Consider Covariance Extended Kalman Filter (CCEKF) implementation can be found in Algorithm 2.2. The reader can find this algorithm by inputting the matrices (2.18) into the equations (2.19), and also in lines 2 and 3 of Algorithm 2.1.

Algorithm 2.2 Consider Covariance Extended Kalman Filter. [42, 44]

```
Require: Prior state $\hat{\mathbf{x}}_k$ and covariance matrices $\{\mathbf{P}_k^{\mathbf{xx}}, \mathbf{P}_k^{\mathbf{xc}}\}$ estimates obtained in the previous temporal step given observations up to, $\mathcal{Y}_{1:k}$, that will be corrected using the observation $\mathbf{y}_{k+1}$ of the current time step and the estimates $\left\{\hat{\mathbf{c}}_k, \mathbf{P}_k^{\mathbf{cc}}, \hat{\mathbf{c}}_{k+1}, \mathbf{P}_{k+1}^{\mathbf{cc}}\right\}$ externally computed.
Ensure: Novel state $\hat{\mathbf{x}}_{k+1}$ and covariance $\left\{\mathbf{P}_{k+1}^{\mathbf{xx}}, \mathbf{P}_{k+1}^{\mathbf{xc}}\right\}$ estimates.
Algorithm: CCEKF$\left(\hat{\mathbf{x}}_k, \hat{\mathbf{c}}_k, \hat{\mathbf{c}}_{k+1}, \mathbf{P}_k^{\mathbf{xx}}, \mathbf{P}_k^{\mathbf{xc}}, \mathbf{P}_k^{\mathbf{cc}}, \mathbf{P}_{k+1}^{\mathbf{cc}}, \mathbf{y}_{k+1}\right)$ return $\hat{\mathbf{x}}_{k+1}$, $\mathbf{P}_{k+1}^{\mathbf{xx}}$, $\mathbf{P}_{k+1}^{\mathbf{xc}}$
  1: $\hat{\mathbf{x}}_{k+1|k} = \mathbf{f}_{\mathbf{x}}\left(\hat{\mathbf{x}}_k\right) + \mathbf{f}_{\mathbf{c}}\left(\hat{\mathbf{c}}_k\right), \quad \mathbf{F}_k^{\mathbf{x}} = \frac{\mathrm{d}\mathbf{f}_{\mathbf{x}}(\hat{\mathbf{x}}_k)}{\mathrm{d}\mathbf{x}}, \quad \mathbf{F}_k^{\mathbf{c}} = \frac{\mathrm{d}\mathbf{f}_{\mathbf{c}}(\hat{\mathbf{c}}_k)}{\mathrm{d}\mathbf{c}}$    ▷ Prediction:
  2: $\mathbf{P}_{k+1|k}^{\mathbf{xx}} = \mathbf{F}_k^{\mathbf{x}}\mathbf{P}_k^{\mathbf{xx}}\left(\mathbf{F}_k^{\mathbf{x}}\right)^T + \mathbf{F}_k^{\mathbf{c}}\mathbf{P}_k^{\mathbf{cx}}\left(\mathbf{F}_k^{\mathbf{x}}\right)^T + \mathbf{F}_k^{\mathbf{x}}\mathbf{P}_k^{\mathbf{xc}}\left(\mathbf{F}_k^{\mathbf{c}}\right)^T + \mathbf{F}_k^{\mathbf{c}}\mathbf{P}_k^{\mathbf{cc}}\left(\mathbf{F}_k^{\mathbf{c}}\right)^T + \mathbf{Q}$
  3: $\mathbf{P}_{k+1|k}^{\mathbf{xc}} = \mathbf{F}_k^{\mathbf{x}}\mathbf{P}_k^{\mathbf{xc}} + \mathbf{F}_k^{\mathbf{c}}\mathbf{P}_k^{\mathbf{cc}}, \quad \mathbf{P}_{k+1|k}^{\mathbf{cx}} = \left(\mathbf{P}_{k+1|k}^{\mathbf{xc}}\right)^T$
  4: $\mathbf{K}_{k+1}^{\mathbf{x}} = \left[\mathbf{P}_{k+1|k}^{\mathbf{xx}}\left(\mathbf{H}_{k+1}^{\mathbf{x}}\right)^T + \mathbf{P}_{k+1|k}^{\mathbf{xc}}\left(\mathbf{H}_{k+1}^{\mathbf{c}}\right)^T\right] \times$    ▷ Kalman Gain
     $\times \left[\mathbf{H}_{k+1}^{\mathbf{x}}\mathbf{P}_{k+1|k}^{\mathbf{xx}}\left(\mathbf{H}_{k+1}^{\mathbf{x}}\right)^T + \mathbf{H}_{k+1}^{\mathbf{c}}\mathbf{P}_{k+1|k}^{\mathbf{cx}}\left(\mathbf{H}_{k+1}^{\mathbf{x}}\right)^T + \mathbf{H}_{k+1}^{\mathbf{x}}\mathbf{P}_{k+1|k}^{\mathbf{xc}}\left(\mathbf{H}_{k+1}^{\mathbf{c}}\right)^T + \mathbf{H}_{k+1}^{\mathbf{c}}\mathbf{P}_{k+1}^{\mathbf{cc}}\left(\mathbf{H}_{k+1}^{\mathbf{c}}\right)^T + \mathbf{R}\right]^{-1}$
  5: $\hat{\mathbf{x}}_{k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}^{\mathbf{x}}\left[\mathbf{y}_{k+1} - \mathbf{h}_{\mathbf{x}}\left(\hat{\mathbf{x}}_{k+1|k}\right) - \mathbf{h}_{\mathbf{c}}\left(\hat{\mathbf{c}}_{k+1}\right)\right]$    ▷ Correction:
  6: $\mathbf{P}_{k+1}^{\mathbf{xx}} = \left(\mathbf{I} - \mathbf{K}_{k+1}^{\mathbf{x}}\mathbf{H}_{k+1}^{\mathbf{x}}\right)\mathbf{P}_{k+1|k}^{\mathbf{xx}} - \mathbf{K}_{k+1}^{\mathbf{x}}\mathbf{H}_{k+1}^{\mathbf{c}}\mathbf{P}_{k+1|k}^{\mathbf{cx}}, \quad \mathbf{H}_{k+1}^{\mathbf{x}} = \frac{\mathrm{d}\mathbf{h}_{\mathbf{x}}\left(\hat{\mathbf{x}}_{k+1|k}\right)}{\mathrm{d}\mathbf{x}}$
  7: $\mathbf{P}_{k+1}^{\mathbf{xc}} = \left(\mathbf{I} - \mathbf{K}_{k+1}^{\mathbf{x}}\mathbf{H}_{k+1}^{\mathbf{x}}\right)\mathbf{P}_{k+1|k}^{\mathbf{xc}} - \mathbf{K}_{k+1}^{\mathbf{x}}\mathbf{H}_{k+1}^{\mathbf{c}}\mathbf{P}_{k+1}^{\mathbf{cc}}, \quad \mathbf{H}_{k+1}^{\mathbf{c}} = \frac{\mathrm{d}\mathbf{h}_{\mathbf{c}}(\hat{\mathbf{c}}_{k+1})}{\mathrm{d}\mathbf{c}}$
  8: return $\hat{\mathbf{x}}_{k+1}$, $\mathbf{P}_{k+1}^{\mathbf{xx}}$, $\mathbf{P}_{k+1}^{\mathbf{xc}}$
```

In this estimation tool, it is assumed that the consider parameters are constant throughout one iteration in the sense that they will not be estimated and that their *a priori* estimated and associated covariance matrix is known. In conclusion, a consider covariance implementation in a filtering procedure attempts to quantify the effects of [44]: 1) Non-estimated parameters, $\mathbf{c}$, whose uncertainty would otherwise be neglected in the estimation procedure; 2) Incorrect *a priori* covariance for the *a priori* estimate of $\mathbf{x}$; 3) Incorrect *a priori* covariance for the measurement noise.

2.3.3 Consensus Filter

Consensus filtering has become a cornerstone within the realm of robotics for enabling a group of agents to converge to a global consensus estimate. The integration of the Consensus Filter (CF) with navigational systems is pivotal for the decentralization of formations and computational power distribution, empowering individual robots to process and share their estimations with neighboring agents.

The attainment of a consensus state typically unfolds through iterative average consensus methods. During these iterations, agents engage in local averaging processes on a specified state. A seminal strategy for combining the CF with the KF, mindful of communication constraints, involves the synchronization of state estimates by comparing the i-th agent's state prediction $\hat{\mathbf{x}}_{k+1}^{(i)}$ with those of its in-neighbors $\forall j \in \mathcal{N}_i : \hat{\mathbf{x}}_{k+1|k}^{(j)}$, as [17, 26]:

$$\hat{\mathbf{x}}_{k+1}^{(i)} = \hat{\mathbf{x}}_{k+1|k}^{(i)} + \mathbf{K}_{k+1}^{(i)} \left[\mathbf{y}_{k+1}^{(i)} - \mathbf{h}_i \left(\hat{\mathbf{x}}_{k+1|k}^{(i)} \right) \right] + \gamma \frac{\mathbf{P}_{k+1|k}^{(i)}}{1 + \left\| \mathbf{P}_{k+1|k}^{(i)} \right\|_F} \sum_{j \in \mathcal{N}_i} \left(\hat{\mathbf{x}}_{k+1|k}^{(j)} - \hat{\mathbf{x}}_{k+1|k}^{(i)} \right), \tag{2.20}$$

thus honing a more refined state estimate $\hat{\mathbf{x}}_{k+1}^{(i)}$. Here, $\mathcal{N}_i$ denotes the in-neighborhood for the i-th node, that is, all the node which can undirectly communicate with the latter, $\| \cdot \|_F$ symbolizes the Frobenius norm, while $\gamma > 0$ represents a user-defined constant that indicates the degree of reliance on the consensus-based estimate.

This rudimentary CF implementation is recognized as a less-than-ideal solution, heavily reliant on approximations within covariance update protocols, specifically concerning the intricacies associated with solving Riccati equations. This method has subsequently been superseded by a CI strategy [26]. Through recurrent consensus cycles, agents collectively converge towards a locally averaged consensus state [16], a topic that will receive further elucidation in Chapter 3.

2.3.4 Diffusion Techniques

Adaptive networks consist of a collection of spatially distributed nodes that are linked together through a topology and that cooperate through local interactions. Adaptive networks are well-suited to perform decentralized information processing and inference tasks and to model complex and self-organized behavior encountered in biological systems.

For instance, we can identify two main categories of fully decentralized methods: consensus strategies and diffusion strategies. Original implementations of the consensus strategy relied on the use of two timescales: one timescale for the collection of measurements across the nodes and another timescale to iterate sufficiently enough over the collected data to attain agreement before the process is repeated. Unfortunately, two timescale implementations hinder the ability to perform real-time recursive estimation and adaptation when measurement data keeps streaming in and the network is of high complexity. For this reason, in this subsection, we focus instead on consensus implementations that operate in a single timescale. Such implementations appear in several recent works and are largely motivated by the procedure developed earlier for the solution of distributed optimization problems. [45]

The main motivation for the introduction of diffusion strategies in this work was the desire to develop distributed schemes that can respond in real-time to the continuous streaming of data at the nodes by operating over a single timescale. Since their inception, diffusion strategies have been applied to model various forms of complex behavior encountered in nature; they have also been adopted to solve distributed optimization problems advantageously; and have been studied under varied conditions. Diffusion strategies are inherently single-timescale implementations and are therefore naturally amenable to real-time and recursive implementations. It turns out that the dynamics of the consensus and diffusion strategies differ in important ways, which in turn impact the mean-square behavior of the respective networks in a fundamental manner. [45]

The best known diffusion strategies are Adapt Then Combine (ATC) and Combine Then Adapt (CTA) strategies [30], both represented by the following expressions

$$
\begin{array}{l|l}
\boldsymbol{\psi}^{(i)}_{k+1} = \hat{\mathbf{x}}^{(i)}_{k+1|k} + \sum_{j\in\mathcal{N}_i} \mathbf{K}^{(j)}_{k+1}\left[\mathbf{y}^{(j)}_{k+1} - \mathbf{h}_j\left(\hat{\mathbf{x}}^{(i)}_{k+1|k}\right)\right] & \boldsymbol{\psi}^{(i)}_{k+1} = \sum_{j\in\mathcal{N}_i} a_{ij}\hat{\mathbf{x}}^{(j)}_{k+1|k} \qquad \boxed{\text{CTA}} \\
\hat{\mathbf{x}}^{(i)}_{k+1} = \sum_{j\in\mathcal{N}_i} a_{ij}\boldsymbol{\psi}^{(j)}_{k+1} \qquad \boxed{\text{ATC}} & \hat{\mathbf{x}}^{(i)}_{k+1} = \boldsymbol{\psi}^{(i)}_{k+1} + \sum_{j\in\mathcal{N}_i} \mathbf{K}^{(j)}_{k+1}\left[\mathbf{y}^{(j)}_{k+1} - \mathbf{h}_j\left(\boldsymbol{\psi}^{(i)}_{k+1}\right)\right]
\end{array}
$$

where a_{ij} are the elements of the combination matrix $\mathbf{A}$, which sums to one in every row. The ATC strategy consists of an incremental step followed by a diffusion step, while the CTA strategy consists of the reverse.

The analysis in [45] confirms that under constant step sizes, diffusion strategies allow information to diffuse more thoroughly through networks and this property has a favorable effect on the evolution of the network. It shows that diffusion networks converge faster and reach a lower mean-square deviation than consensus networks, and their mean-square stability is insensitive to the choice of the combination weights. In comparison, and surprisingly, it also shows that consensus networks can become unstable even if all the individual nodes are stable and able to solve estimation tasks on their own. In other words, the learning curve of a cooperative consensus network can diverge even if the learning curves for the non-cooperative individual nodes converge. When this occurs, cooperation over the network leads to a catastrophic failure of the estimation task. This behavior does not occur for diffusion networks: the stability of the individual nodes is sufficient to ensure the stability of the diffusion network regardless of the combination weights. The analysis also suggests that diffusion strategies provide a proper way to enforce cooperation over networks; their operation is such that diffusion networks will always remain stable irrespective of the combination topology.

Due to time constraints, these strategies haven't been included in this report. However, they will be further explored in the upcoming Master's Thesis document.

Chapter 3

Implementation

Exploration is wired into our brains. If we can see the horizon, we want to know what's beyond.

Buzz Aldrin

The objective of this chapter is to state the problem to solve and explain the implementation of the algorithms presented in the previous Chapter, how to specifically apply them to this problem and some particular features they have. In the following sections, the dynamics and observations modeling are presented for the various types of spacecraft. The algorithms presented here will be used in Chapter 4 to perform the simulations.

3.1 Problem Statement

In this work, we will consider an LEO-satellite FF that follows a chief/deputy coordination approach, where one spacecraft (the chief) follows a reference orbit, while others (deputies) adjust their positions relative to the chief. This network can similarly be represented by an undirected communication graph of nodes and arcs $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where it is assumed that each satellite in the network has processing and communication capabilities: each satellite has the ability to process their state estimations as well as exchanging information with their in-neighboring spacecraft. Following a chief/deputy topology, the set of nodes can also be represented by $\mathcal{N} = \mathcal{C} \cup \mathcal{D}$, where $\mathcal{C}$ is the set of chiefs that have access to GPS measurements and only transmit their estimates to their in-neighboring deputies. The deputies are represented by the set $\mathcal{D}$ and only have access to range measurements and communication between all nodes. $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ is the set of arcs (connections) between nodes in the formation such that $(i, j) \in \mathcal{A}$ if node j can receive data from node i. The in-neighborhood of the i-th satellite is represented by $\mathcal{N}_i \triangleq \{j : (j, i) \in \mathcal{A}\}$.

The goal here is to accurately estimate the state vector for every satellite in the FF in an ECI frame, that is, $\forall i \in \mathcal{N} : \mathbf{x}_i = \begin{bmatrix} \mathbf{r}_i^T & \dot{\mathbf{r}}_i^T \end{bmatrix}^T$, the concatenation of spatial position and velocity vectors, which has $\dim(\mathbf{x}_i) = 6 \times 1$, using the algorithms introduced in Section 2.3.

3.1.1 Dynamics Modelling

As it was previously mentioned in Chapter 2, for LEO orbits the main contributions to the acceleration of the spacecraft are the Earth's gravitational attraction (2.1), and the J_2 zonal harmonic for the Earth's oblateness (2.6) and atmospheric drag (2.8) orbital perturbations.

Therefore, for the i-th satellite in the FF, the dynamics of the spacecraft can be described by the following:

$$\dot{\mathbf{x}}_i = \mathbf{f}(\mathbf{x}_i) = \begin{bmatrix} \dot{\mathbf{r}}_i \\ \ddot{\mathbf{r}}_{\text{grav}}^{(i)} + \ddot{\mathbf{r}}_{J_2}^{(i)} + \ddot{\mathbf{r}}_{\text{drag}}^{(i)} \end{bmatrix} \tag{3.1}$$

where all satellites are considered to be identical and therefore have the same dynamical model. Noting that the dynamics are defined in the continuous domain, numerically integrating (3.1) from the past time step t_k to the current time step t_{k+1} (time step $\Delta t = t_{k+1} - t_k$) gives an *a priori* state estimate through [46]

$$\hat{\mathbf{x}}^{(i)}_{k+1|k} = \hat{\mathbf{x}}^{(i)}_{k} + \int_{t_k}^{t_{k+1}} \mathbf{f}\left(\hat{\mathbf{x}}^{(i)}_{k}\right) \, \mathrm{d}t \tag{3.2}$$

which can be solved by a numerical integration method like the classical Runge-Kutta.

Due to the errors in modeling the true dynamics of the system, this propagation will accordingly introduce some error into the state estimate, which must be quantified in the error covariance matrix $\mathbf{P}^{(i)}_{k+1}$. Contrasting the nonlinear propagation of the state estimate, the error covariance is propagated using a linearized version of the dynamics. The process of linearization involves evaluating the Jacobian of $\mathbf{f}(\cdot)$ at the best state estimate from the previous time step, namely the *a posteriori* state estimate $\hat{\mathbf{x}}^{(i)}_{k}$ whereby the Jacobian is

$$\mathbf{F}^{(i)}_{k} = \frac{\mathrm{d}\mathbf{f}\left(\hat{\mathbf{x}}^{(i)}_{k}\right)}{\mathrm{d}\mathbf{x}_i}, \quad \text{where} \quad \frac{\mathrm{d}\mathbf{f}(\mathbf{x}_i)}{\mathrm{d}\mathbf{x}_i} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_3 \\ \frac{\partial \ddot{\mathbf{r}}^{(i)}_{\text{grav}}}{\partial \mathbf{r}_i} + \frac{\partial \ddot{\mathbf{r}}^{(i)}_{J_2}}{\partial \mathbf{r}_i} + \frac{\partial \ddot{\mathbf{r}}^{(i)}_{\text{drag}}}{\partial \mathbf{r}_i} & \frac{\partial \ddot{\mathbf{r}}^{(i)}_{\text{drag}}}{\partial \dot{\mathbf{r}}_i} \end{bmatrix}, \tag{3.3}$$

which can be analytically obtained from the sets of equations {(2.3), (2.7), (2.11)} for position partial derivatives and (2.10) for velocity partial derivatives.

The classic Runge-Kutta (RK4) method [35] can be used to perform the integration from the previous iteration to the next with an intermediate propagation time step. The same procedure was performed to take continuous time jacobians and convert them to RK4 discrete jacobians for the dynamics model, which can be seen in Algorithm 3.1, where the i indentificator was dropped for simplification.

Algorithm 3.1 Runge-Kutta (4th order). [35]

```
Require: Initial state vector $\mathbf{x}_k$, the system's dynamics function $\mathbf{f}(\cdot)$ and the time step $\Delta t$.
Ensure: Next state vector $\mathbf{x}_{k+1}$ and the Jacobian matrix $\mathbf{F}_k$.
Algorithm: RK4$(\mathbf{x}_k, \mathbf{f}(\cdot), \Delta t)$ return $\mathbf{x}_{k+1}$, $\mathbf{F}_k$
  $\mathbf{k}_1 = \mathbf{f}(\mathbf{x}_k)$
  $\mathbf{k}_2 = \mathbf{f}\left(\mathbf{x}_k + \frac{\Delta t}{2}\mathbf{k}_1\right)$
  $\mathbf{k}_3 = \mathbf{f}\left(\mathbf{x}_k + \frac{\Delta t}{2}\mathbf{k}_2\right)$
  $\mathbf{k}_4 = \mathbf{f}(\mathbf{x}_k + \Delta t\mathbf{k}_3)$
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\Delta t}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$    ▷ New state vector
  $\frac{\mathrm{d}\mathbf{k}_1}{\mathrm{d}\mathbf{x}_k} = \frac{\mathrm{d}\mathbf{f}(\mathbf{x}_k)}{\mathrm{d}\mathbf{x}_k}$    ▷ The following four derivatives we computed using the chain rule
  $\frac{\mathrm{d}\mathbf{k}_2}{\mathrm{d}\mathbf{x}_k} = \frac{\mathrm{d}\mathbf{f}\left(\mathbf{x}_k + \frac{\Delta t}{2}\mathbf{k}_1\right)}{\mathrm{d}\mathbf{x}_k}\left(\mathbf{I}_6 + \frac{\Delta t}{2}\frac{\mathrm{d}\mathbf{k}_1}{\mathrm{d}\mathbf{x}_k}\right)$
  $\frac{\mathrm{d}\mathbf{k}_3}{\mathrm{d}\mathbf{x}_k} = \frac{\mathrm{d}\mathbf{f}\left(\mathbf{x}_k + \frac{\Delta t}{2}\mathbf{k}_2\right)}{\mathrm{d}\mathbf{x}_k}\left(\mathbf{I}_6 + \frac{\Delta t}{2}\frac{\mathrm{d}\mathbf{k}_2}{\mathrm{d}\mathbf{x}_k}\right)$
  $\frac{\mathrm{d}\mathbf{k}_4}{\mathrm{d}\mathbf{x}_k} = \frac{\mathrm{d}\mathbf{f}(\mathbf{x}_k + \Delta t\mathbf{k}_3)}{\mathrm{d}\mathbf{x}_k}\left(\mathbf{I}_6 + \Delta t\frac{\mathrm{d}\mathbf{k}_3}{\mathrm{d}\mathbf{x}_k}\right)$
  $\mathbf{F}_k = \mathbf{I}_6 + \frac{\Delta t}{6}\left(\frac{\mathrm{d}\mathbf{k}_1}{\mathrm{d}\mathbf{x}_k} + 2\frac{\mathrm{d}\mathbf{k}_2}{\mathrm{d}\mathbf{x}_k} + 2\frac{\mathrm{d}\mathbf{k}_3}{\mathrm{d}\mathbf{x}_k} + \frac{\mathrm{d}\mathbf{k}_4}{\mathrm{d}\mathbf{x}_k}\right)$    ▷ Dynamics Jacobian matrix
  return $\mathbf{x}_{k+1}$, $\mathbf{F}_k$
```

3.1.2 Observations Modelling

For chiefs, the observations are given by GPS measurements, which is the position of the spacecraft in the ECI frame as follows:

$$\forall i \in \mathcal{C} : \mathbf{y}_i = \mathbf{h}_{\mathcal{C}}(\mathbf{x}_i) = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}, \tag{3.4}$$

the respective Jacobian matrix is given by

$$\forall i \in \mathcal{C} : \mathbf{H}_{k+1}^{(i)} = \frac{\mathrm{d}\mathbf{h}_{\mathcal{C}}\left(\hat{\mathbf{x}}_{k+1|k}^{(i)}\right)}{\mathrm{d}\mathbf{x}_i} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times 3} \end{bmatrix}. \tag{3.5}$$

On the contrary, for deputies, the observations are given by range measurements between them and their in-neighbors. Therefore, the observations model for deputies is given by:

$$\forall i \in \mathcal{D}, \forall j \in \mathcal{N}_i : \mathbf{y}_{ij} = \mathbf{h}_{\mathcal{D}}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{r}_i - \mathbf{r}_j\|_2, \tag{3.6}$$

and the respective Jacobian matrix is given by

$$\forall i \in \mathcal{D}, \forall j \in \mathcal{N}_i : \mathbf{H}_{k+1}^{(i)} = \frac{\partial \mathbf{h}_{\mathcal{D}}\left(\hat{\mathbf{x}}_{k+1|k}^{(i)}, \hat{\mathbf{x}}_{k+1|k}^{(j)}\right)}{\partial \mathbf{x}_i}, \quad \text{where} \quad \frac{\partial \mathbf{h}_{\mathcal{D}}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} = \begin{bmatrix} \frac{\mathbf{r}_i^T - \mathbf{r}_j^T}{\|\mathbf{r}_i - \mathbf{r}_j\|_2} & \mathbf{0}_{1\times 3} \end{bmatrix}. \tag{3.7}$$

Moreover, the partial derivative of a deputy's observation model with respect to the state of any of its in-neighbors is given by

$$\frac{\partial \mathbf{h}_{\mathcal{D}}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_j} = \begin{bmatrix} -\frac{\mathbf{r}_i^T - \mathbf{r}_j^T}{\|\mathbf{r}_i - \mathbf{r}_j\|_2} & \mathbf{0}_{1\times 3}, \end{bmatrix} \tag{3.8}$$

$\forall i \in \mathcal{D}$ and $\forall j \in \mathcal{N}_i$.

3.2 Fully Centralized Extended Kalman Filter

For this solution, there is one central entity estimating one big state vector for all the FF using Algorithm 2.1. This state vector is the concatenation of the state vectors of all the spacecraft in the FF, that is, $\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_N^T \end{bmatrix}^T$, where N is the number of spacecraft in the FF, in other words, the cardinality of $\mathcal{N}$. The state vector has $\dim(\mathbf{x}) = 6N \times 1$.

Firstly, the predicted state vector is obtained by applying Algorithm 3.1 N times, one for each spacecraft in the FF, followed by vertically merging these vectors. Similarly, the dynamics Jacobian is calculated using the same approach. However, instead of a vertical concatenation, it is assembled diagonally, resulting in a sparse structure. In this case, the measurement model makes use of (3.8), since the state is the concatenation of the state vectors of all the spacecraft in the FF.

3.3 Consensus Extended Kalman Filter

Algorithm 3.2 performs the estimation of the information matrix $\mathbf{\Omega}^{(i)}_{k+1}$ and vector $\mathbf{q}^{(i)}_{k}$ for the i-th spacecraft using the HCMCI architecture. These variables relate to the inverse of the covariance matrix and are specifically used in distributed problems as they decrease the complexity of measurement processing. This reduction stems from the additive nature of the information form; it allows for new measurements to be incorporated through straightforward addition. This feature is especially beneficial for efficiently recursively processing multiple sets of measurements. In this solution, each spacecraft has its estimation of the formation state and performs L consensus rounds to improve this estimate from the information between the different formation spacecraft.

Algorithm 3.2 Hybrid Consensus on Measurements and Consensus on Information. [16]

```
Require: Prior state $\hat{\mathbf{x}}^{(i)}_{k}$ and information matrix $\mathbf{\Omega}^{(i)}_{k}$ estimates, $\forall i \in \mathcal{N}$ obtained in the previous temporal step given observations up to time $k$, that will be corrected using the observation $\mathbf{y}^{(i)}_{k+1}$.
Ensure: Novel state $\hat{\mathbf{x}}^{(i)}_{k+1}$ and information matrix $\mathbf{\Omega}^{(i)}_{k+1}$ estimates, $\forall i \in \mathcal{N}$.
Algorithm: HCMCI$\left(\hat{\mathbf{x}}^{(i)}_{k}, \mathbf{y}^{(i)}_{k+1}, \forall i \in \mathcal{N}\right)$ return $\hat{\mathbf{x}}^{(i)}_{k+1}, \forall i \in \mathcal{N}$
for $i \in \mathcal{N}$ do
                                                                    ▷ Prediction
    $\hat{\mathbf{x}}^{(i)}_{k+1|k} = \mathbf{f}\left(\hat{\mathbf{x}}^{(i)}_{k}\right), \quad \mathbf{F}^{(i)}_{k} = \frac{\mathrm{d}\mathbf{f}\left(\hat{\mathbf{x}}^{(i)}_{k}\right)}{\mathrm{d}\mathbf{x}}$ and $\mathbf{H}^{(i)}_{k+1} = \frac{\mathrm{d}\mathbf{h}_i\left(\hat{\mathbf{x}}^{(i)}_{k+1|k}\right)}{\mathrm{d}\mathbf{x}}$
    $\mathbf{\Omega}^{(i)}_{k+1|k} = \mathbf{Q}_i^{-1} - \mathbf{Q}_i^{-1}\mathbf{F}^{(i)}_{k}\left[\mathbf{\Omega}^{(i)}_{k} + \left(\mathbf{F}^{(i)}_{k}\right)^T \mathbf{Q}_i^{-1}\mathbf{F}^{(i)}_{k}\right]^{-1}\left(\mathbf{F}^{(i)}_{k}\right)^T \mathbf{Q}_i^{-1}$
    $\mathbf{q}^{(i)}_{k+1|k} = \mathbf{\Omega}^{(i)}_{k+1|k}\hat{\mathbf{x}}^{(i)}_{k+1|k}$
    $\delta\mathbf{q}^{(i)}_{k+1} = \left(\mathbf{H}^{(i)}_{k+1}\right)^T \mathbf{R}_i^{-1}\left[\mathbf{y}^{(i)}_{k+1} - \mathbf{h}_i\left(\hat{\mathbf{x}}^{(i)}_{k+1|k}\right) + \mathbf{H}^{(i)}_{k+1}\hat{\mathbf{x}}^{(i)}_{k+1|k}\right]$
    $\delta\mathbf{\Omega}^{(i)}_{k+1} = \left(\mathbf{H}^{(i)}_{k+1}\right)^T \mathbf{R}_i^{-1}\mathbf{H}^{(i)}_{k+1}$
                                                                    ▷ Consensus
    $\delta\mathbf{q}^{(i)}_{k+1}(0) = \delta\mathbf{q}^{(i)}_{k+1}, \quad \delta\mathbf{\Omega}^{(i)}_{k+1}(0) = \delta\mathbf{\Omega}^{(i)}_{k+1}$
    $\mathbf{q}^{(i)}_{k+1|k}(0) = \mathbf{q}^{(i)}_{k+1|k}, \quad \mathbf{\Omega}^{(i)}_{k+1|k}(0) = \mathbf{\Omega}^{(i)}_{k+1|k}$
    for $\ell = 0, 1, \ldots, L-1$ do
        $\delta\mathbf{q}^{(i)}_{k+1}(\ell+1) = \sum_{j\in\mathcal{N}_i}\pi_{i,j}\delta\mathbf{q}^{(j)}_{k+1}(\ell), \quad \delta\mathbf{\Omega}^{(i)}_{k+1}(\ell+1) = \sum_{j\in\mathcal{N}_i}\pi_{i,j}\delta\mathbf{\Omega}^{(j)}_{k+1}(\ell)$
        $\mathbf{q}^{(i)}_{k+1|k}(\ell+1) = \sum_{j\in\mathcal{N}_i}\pi_{i,j}\mathbf{q}^{(j)}_{k+1|k}(\ell), \quad \mathbf{\Omega}^{(i)}_{k+1|k}(\ell+1) = \sum_{j\in\mathcal{N}_i}\pi_{i,j}\mathbf{\Omega}^{(j)}_{k+1|k}(\ell)$
    end for
                                                                    ▷ Correction
    $\mathbf{q}^{(i)}_{k+1} = \mathbf{q}^{(i)}_{k+1|k}(L) + \gamma^{(i)}_{k+1}\delta\mathbf{q}^{(i)}_{k+1}(L)$
    $\mathbf{\Omega}^{(i)}_{k+1} = \mathbf{\Omega}^{(i)}_{k+1|k}(L) + \gamma^{(i)}_{k+1}\delta\mathbf{\Omega}^{(i)}_{k+1}(L)$
    $\hat{\mathbf{x}}^{(i)}_{k+1} = \left(\mathbf{\Omega}^{(i)}_{k+1}\right)^{-1}\mathbf{q}^{(i)}_{k+1}$
end for
return $\hat{\mathbf{x}}^{(i)}_{k+1}, \mathbf{\Omega}^{(i)}_{k+1}, \forall i \in \mathcal{N}$
```

In consensus filtering, there is no direct need for a fully connected graph, the consensus rounds correspond to L cycles where the satellites exchange information between them to reach a global consensus on the state. The objective is that if L tends to infinity the filter converges to the results of the centralized approach.

There are two types of consensus weights, $\gamma_{k+1}^{(i)}$ and $\pi_{i,j}$, either one can be optimized to improve accuracy. The first weight is introduced to counteract the possible underweighting of novel information. Kamal [47] proposed $\gamma_{k+1}^{(i)} = N$, which provides a solution for the decentralized algorithm to converge to a centralized solution when L tends to infinity. The second consensus weight defines the importance of each spacecraft in the consensus rounds to the final estimation. The Metropolis Weights [48] provide convergence to the consensus average, where each arc's weight corresponds to

$$\pi_{i,j} = \begin{cases} \frac{1}{1+\max[\deg(i),\deg(j)]}, & \text{if } (i,j) \in \mathcal{A} \\ 1 - \sum_{j\in\mathcal{N}_i} \pi_{i,j}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}, \tag{3.9}$$

where $\deg(i)$ is the cardinality of $\mathcal{N}_i$.

3.4 Consider Covariance Extended Kalman Filter

For this solution, the chiefs just estimate their state only resorting to their GPS measurements, using the Algorithm 2.1. On the other hand, the deputies filter their state, which is the concatenation of their state and the states of their in-neighbor deputies, using the Algorithm 3.3 after relative range measurements to their in-neighbor spacecraft have been taken and after collecting the chiefs' state, the consider parameter, and covariance matrix estimates. Since the dynamics of the chief do not have an impact on the deputies' dynamics, the function $\mathbf{f_c}(\cdot)$ as well as the Jacobian $\mathbf{F}_k^{\mathbf{c}}$ are zero.

Algorithm 3.3 Deputies CCEKF

Require: Prior deputies state $\hat{\mathbf{x}}_k$ and covariance matrices $\{\mathbf{P}_k^{\mathbf{xx}}, \mathbf{P}_k^{\mathbf{xc}}\}$ estimates obtained in the previous temporal step given observations up to, $\mathcal{Y}_{1:k}^{\mathcal{D}}$, that will be corrected using the observation $\mathbf{y}_{k+1}$ of the current time step and the consider parameter estimates externally computed by the chiefs.

Ensure: Novel deputies state $\hat{\mathbf{x}}_{k+1}^{\mathcal{D}}$ and covariance $\left\{\mathbf{P}_{k+1}^{\mathbf{xx}}, \mathbf{P}_{k+1}^{\mathbf{xc}}\right\}$ estimates.

Algorithm: CCEKF$_{\mathcal{D}}$ $\left(\hat{\mathbf{x}}_k^{\mathcal{D}}, \mathbf{P}_k^{\mathbf{xx}}, \mathbf{P}_k^{\mathbf{xc}}, \mathbf{y}_{k+1}^{\mathcal{D}}\right)$ **return** $\hat{\mathbf{x}}_{k+1}^{\mathcal{D}}, \mathbf{P}_{k+1}^{\mathbf{xx}}, \mathbf{P}_{k+1}^{\mathbf{xc}}$

```
$\hat{\mathbf{x}}_{k+1|k}^{\mathcal{D}} = \mathbf{f_x}\left(\hat{\mathbf{x}}_k^{\mathcal{D}}\right), \quad \mathbf{F}_k^{\mathbf{x}} = \frac{\mathrm{d}\mathbf{f_x}\left(\hat{\mathbf{x}}_k^{\mathcal{D}}\right)}{\mathrm{d}\mathbf{x}}$    ▷ Prediction:
$\mathbf{P}_{k+1|k}^{\mathbf{xx}} = \mathbf{F}_k^{\mathbf{x}}\mathbf{P}_k^{\mathbf{xx}}\left(\mathbf{F}_k^{\mathbf{x}}\right)^T + \mathbf{Q}$
$\mathbf{P}_{k+1|k}^{\mathbf{xc}} = \mathbf{F}_k^{\mathbf{x}}\mathbf{P}_k^{\mathbf{xc}}, \quad \mathbf{P}_{k+1|k}^{\mathbf{cx}} = \left(\mathbf{P}_{k+1|k}^{\mathbf{xc}}\right)^T$
Get $\hat{\mathbf{c}}_{k+1} = \hat{\mathbf{x}}_{k+1}^{\mathcal{C}}$ and $\mathbf{P}_{k+1}^{\mathbf{cc}} = \mathbf{P}_{k+1}^{\mathcal{C}}$ from chief filter.    ▷ Communication with the chiefs.
$\mathbf{K}_{k+1}^{\mathbf{x}} = \{$exactly as in Algorithm 2.2$\}$    ▷ Kalman Gain
$\hat{\mathbf{x}}_{k+1}^{\mathcal{D}} = \hat{\mathbf{x}}_{k+1|k}^{\mathcal{D}} + \mathbf{K}_{k+1}^{\mathbf{x}}\left[\mathbf{y}_{k+1}^{\mathcal{D}} - \mathbf{h_x}\left(\hat{\mathbf{x}}_{k+1|k}^{\mathcal{D}}\right) - \mathbf{h_c}\left(\hat{\mathbf{c}}_{k+1}\right)\right]$    ▷ Correction:
$\mathbf{P}_{k+1}^{\mathbf{xx}} = \left(\mathbf{I} - \mathbf{K}_{k+1}^{\mathbf{x}}\mathbf{H}_{k+1}^{\mathbf{x}}\right)\mathbf{P}_{k+1|k}^{\mathbf{xx}} - \mathbf{K}_{k+1}^{\mathbf{x}}\mathbf{H}_{k+1}^{\mathbf{c}}\mathbf{P}_{k+1|k}^{\mathbf{cx}}, \quad \mathbf{H}_{k+1}^{\mathbf{x}} = \frac{\mathrm{d}\mathbf{h_x}\left(\hat{\mathbf{x}}_{k+1|k}^{\mathcal{D}}\right)}{\mathrm{d}\mathbf{x}}$
$\mathbf{P}_{k+1}^{\mathbf{xc}} = \left(\mathbf{I} - \mathbf{K}_{k+1}^{\mathbf{x}}\mathbf{H}_{k+1}^{\mathbf{x}}\right)\mathbf{P}_{k+1|k}^{\mathbf{xc}} - \mathbf{K}_{k+1}^{\mathbf{x}}\mathbf{H}_{k+1}^{\mathbf{c}}\mathbf{P}_{k+1}^{\mathbf{cc}}, \quad \mathbf{H}_{k+1}^{\mathbf{c}} = \frac{\mathrm{d}\mathbf{h_c}\left(\hat{\mathbf{c}}_{k+1}\right)}{\mathrm{d}\mathbf{c}}$
return $\hat{\mathbf{x}}_{k+1}^{\mathcal{D}}, \mathbf{P}_{k+1}^{\mathbf{xx}}, \mathbf{P}_{k+1}^{\mathbf{xc}}$
```

Chapter 4

Preliminary Results

Mystery creates wonder and wonder is the basis of man’s desire to understand.

Neil Armstrong

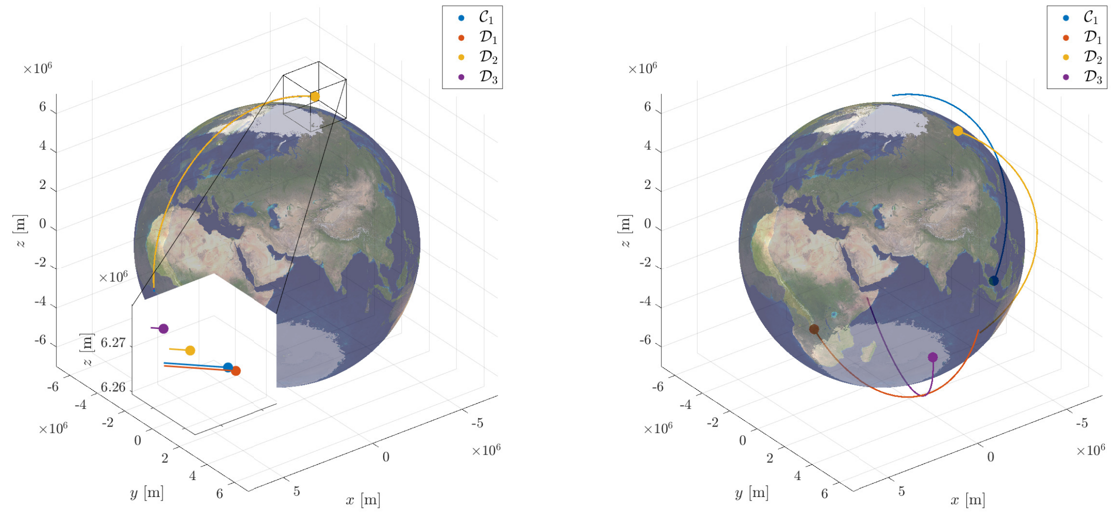
Two distinct formation configurations will be evaluated. Formation I, inspired by the prior V-R3x mission, features spacecraft trajectories detailed in Table 4.1 and Figure 4.1a, which maintain a maximum relative distance of approximately $10\,\mathrm{km}$. Formation II, as showcased in Table 4.2 and Figure 4.1b, involves orbits distinguished by markedly varied classical orbital elements. This work assumes uninterrupted communication among all satellites, regardless of their actual visibility or range limitations. These considerations will be addressed in future work.

Table 4.1: Classical Orbital Elements for the Formation I.

Satellite	i [°]	Ω [°]	ϖ [°]	a [km]	e	θ_0 [°]
Chief 1 ($\mathcal{C}_1$)	97.49	0	0	6903.50	0.0011	0
Deputy 1 ($\mathcal{D}_1$)	97.49	0	9.23	6903.98	0.0012	350.76
Deputy 2 ($\mathcal{D}_2$)	97.47	0	327.27	6902.67	0.0012	32.72
Deputy 3 ($\mathcal{D}_3$)	97.52	0	330.47	6904.34	0.0014	29.52

Table 4.2: Classical Orbital Elements for the Formation II.

Satellite	i [°]	Ω [°]	ϖ [°]	a [km]	e	θ_0 [°]
Chief 1 ($\mathcal{C}_1$)	97.79	1.5×10^{-5}	303.34	6978	2.6×10^{-6}	157.36
Deputy 1 ($\mathcal{D}_1$)	97.26	272.80	281.15	6978	6.48×10^{-3}	269.52
Deputy 2 ($\mathcal{D}_2$)	97.79	149.99	104.07	6978	6.6×10^{-5}	206.00
Deputy 3 ($\mathcal{D}_3$)	97.79	70	257.43	6978	1.3×10^{-5}	332.57



(a) Formation I.

(b) Formation II.

Figure 4.1: Plot of each simulated satellite formation at $t = 50\,\mathrm{min}$ after orbital initial conditions.

Simulations will serve to verify the feasibility of the three proposed solutions for small-satellite localization systems. These simulations will employ the Monte-Carlo approach, utilizing $M = 100$ samples. In each simulation, the initial position estimates will be randomly offset from the actual starting location, varying in direction for every run. Additionally, a random velocity deviation, corresponding to the state uncertainty of this component, will be incorporated. The duration of each simulation will approximate a real-time span of about $T = 6\,\mathrm{h}$, which translates to roughly four to five complete orbital revolutions.

4.1 Parameters and Metrics

Each formation will consist of $N = 4$ spacecraft, 1 chief and 3 deputies, modeled by $C_D = 2.22$ drag coefficient, a mass of $m = 1\,\mathrm{kg}$ and a frontal area of $A_\perp = 1\,\mathrm{cm}^2$, and will be simulated for $T = 360\,\mathrm{min}$, with a time step of $\Delta t = 60\,\mathrm{s}$, in $M = 100$ Monte-Carlo runs. The Earth was simulated using the following parameters: $J_2 = 0.00108262545$ for its oblateness (second zonal harmonic), $R_\oplus = 6378.1363\,\mathrm{m}$ as its radius and $\mu_\oplus = 3.986 \times 10^5\,\mathrm{km}^3\,\mathrm{s}^{-2}$ as its gravitational parameter.

The chosen process and measurement noise covariance matrices, respectively $\mathbf{Q} = \mathrm{diag}\left(q_x^2, q_y^2, q_z^2, q_{\dot{x}}^2, q_{\dot{y}}^2, q_{\dot{z}}^2\right)$ and $\mathbf{R}_{\mathrm{GPS}} = \mathrm{diag}\left(r_{\mathrm{GPS}}^2, r_{\mathrm{GPS}}^2, r_{\mathrm{GPS}}^2\right)$ or $\mathbf{R}_\leftrightarrow = \mathrm{diag}\left(r_\leftrightarrow^2, r_\leftrightarrow^2, r_\leftrightarrow^2\right)$ (each deputy takes $N-1 = 3$ relative-range measurements), are represented in the Table 4.3.

Table 4.3: Process and measurement noise covariance matrix entries (based on [27]).

Standard Deviation	$\mathcal{C}_1$	$\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$	GPS	Relative-range
q_x, q_y, q_z [km]	1×10^{-4}	1×10^{-3}		
$q_{\dot{x}}, q_{\dot{y}}, q_{\dot{z}}$ [km s^{-1}]	1×10^{-5}	1×10^{-5}		
r_{GPS} [km]			1×10^{-4}	
$r_\leftrightarrow$ [km]				1×10^{-3}

For the following simulations, the initial state estimate is set to the true state plus random fluctuations. This initial variation is characterized by a standard deviation of $100\,\mathrm{m}$ for position and $1\,\mathrm{m}\,\mathrm{s}^{-1}$ for velocity. The initial state's covariance is a diagonal matrix, where the diagonal elements are the squared values of these initial deviations.

The performance metric used will be the mean Root-Mean-Square Error (RMSE) of the position estimates for each Monte-Carlo simulation, mathematically defined as

$$\bar{\varepsilon}_{\mathrm{RMSE}}^{(i)} = \frac{1}{M}\sum_{m=1}^{M} \varepsilon_{\mathrm{RMSE}_m}^{(i)}, \quad \varepsilon_{\mathrm{RMSE}_m}^{(i)} = \sqrt{\frac{1}{T+1-n}\sum_{k=n}^{N}\left(\varepsilon_{k,m}^{(i)}\right)^2}, \quad \varepsilon_{k,m}^{(i)} = \left\|\hat{\mathbf{r}}_{k,m}^{(i)} - \mathbf{r}_{k,m}^{(i)}\right\|_2 \tag{4.1}$$

where a burn-in period of $n = 300$ time steps was used.

4.2 Accuracy Analysis

The set of simulations was performed for each formation, with each filter, and the results for the average absolute deviation and average RMSE for each satellite are presented in Figures 4.2 and 4.3 and Table 4.4, respectively.

We observed that all algorithms successfully converged for each spacecraft, despite being based on linear approximations of a highly non-linear model. If divergence had occurred in Formation I, it might have been attributable to the presence of ambiguous orbits. In Formation I, the simulation starts immedi-

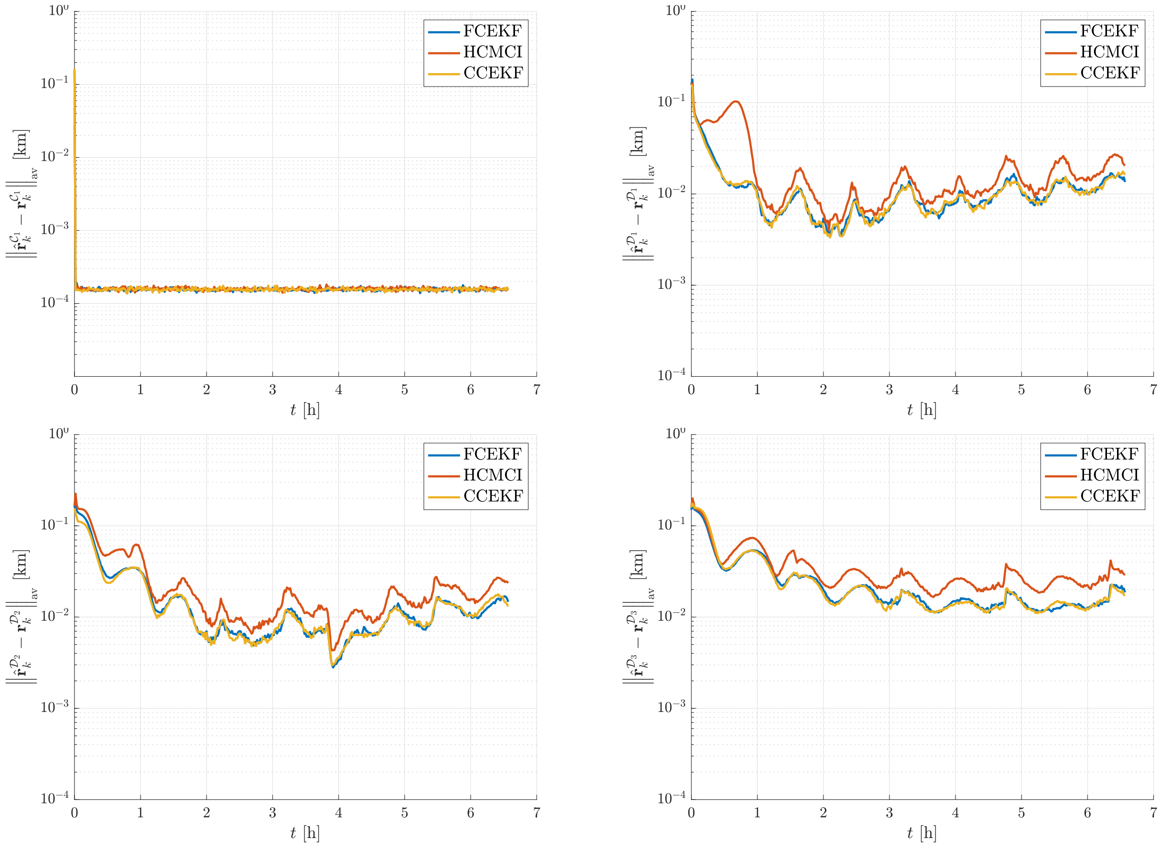


Figure 4.2: Average position absolute error for Formation I for $M = 100$ Monte-Carlo simulations.

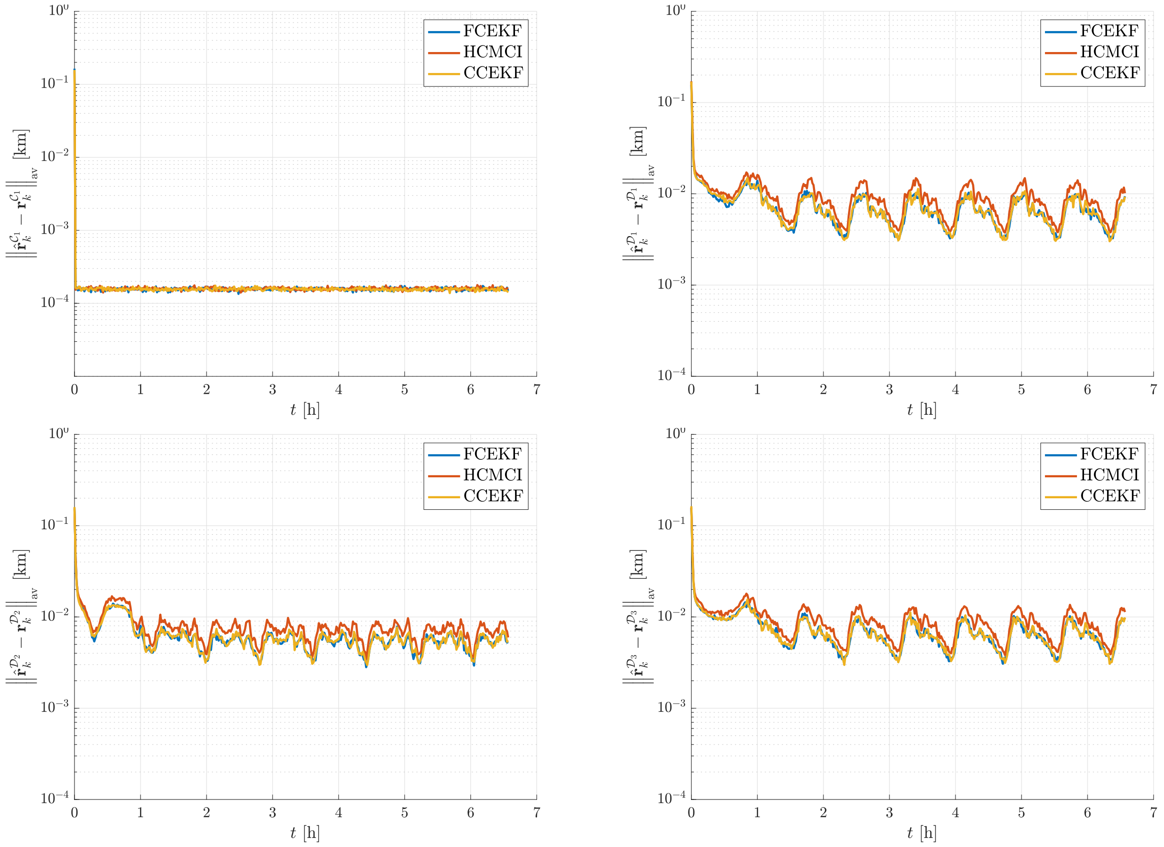


Figure 4.3: Average position absolute error for Formation II for $M = 100$ Monte-Carlo simulations.

Table 4.4: Average RMSE for the two formations over $M = 100$ Monte-Carlo simulations.

Formation	**Filter**	$\bar{\varepsilon}^{\mathcal{C}_1}_{\mathrm{RMSE}}$ [m]	$\bar{\varepsilon}^{\mathcal{D}_1}_{\mathrm{RMSE}}$ [m]	$\bar{\varepsilon}^{\mathcal{D}_2}_{\mathrm{RMSE}}$ [m]	$\bar{\varepsilon}^{\mathcal{D}_3}_{\mathrm{RMSE}}$ [m]
I	FCEKF	0.169	13.983	13.925	16.607
	HCMCI	0.172	21.243	22.515	29.403
	CCEKF	0.169	13.986	14.249	16.369
II	FCEKF	0.170	7.590	6.138	7.378
	HCMCI	0.173	10.183	8.152	9.732
	CCEKF	0.168	7.630	6.232	7.400

ately after the spacecraft leave the transportation container. This phase is marked by limited observability, as the satellites begin in the same position. Their only initial differences are in angular and linear velocities, influenced by their method of container egress. However, the convergence in an ambiguous orbit situation can be explained in various ways. Notably, maintaining a very tight formation with minimal relative distance, and starting from nearly identical absolute positions, could justify this occurrence. In our case, this did not apply.

It was evident that the FCEKF outperformed other methods in terms of convergence speed and RMSE. This result was anticipated, given that the FCEKF accounts for the cross-correlations between all spacecraft within the same model. All algorithms demonstrated comparable accuracy, with the CCEKF ranking as the second-best.

The conducted accuracy tests revealed that the algorithms are more effective in formations with varied classical orbital elements. The results were notably better for Formation II compared to Formation I. Beyond accuracy, when comparing the CCEKF with the HCMCI algorithm, the primary advantage lies in communication efficiencies, which will be addressed in the future.

Bibliography

[1] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano. Consensus-based linear and nonlinear filtering. *IEEE Transactions on Automatic Control*, 60(5):1410–1415, May 2015. ISSN 1558-2523. doi: 10.1109/tac.2014.2357135.

[2] K. Alfriend, S. R. Vadali, P. Gurfil, J. How, and L. Breger. *Spacecraft Formation Flying: Dynamics, Control and Navigation*. Elsevier Science & Technology Books, 2009. ISBN 978-0-08-055965-0.

[3] G.-P. Liu and S. Zhang. A survey on formation control of small satellites. *Proceedings of the IEEE*, 106(3): 440–457, March 2018. ISSN 0018-9219. doi: 10.1109/JPROC.2018.2794879.

[4] S. Nag, C. K. Gatebe, D. W. Miller, and O. L. De Weck. Effect of satellite formations and imaging modes on global albedo estimation. *Acta Astronautica*, 126:77–97, 2016.

[5] G. D. Mauro, M. Lawn, and R. Bevilacqua. Survey on guidance navigation and control requirements for spacecraft formation-flying missions. *Journal of Guidance, Control, and Dynamics*, 41(3):581–602, 2018. doi: 10.2514/1.G002868.

[6] J. R. Wright et al. Optimal orbit determination. *Advances in the Astronautical Sciences*, 112:1123–1134, 2002.

[7] J. R. Vetter. Fifty years of orbit determination. *Johns Hopkins APL technical digest*, 27(3):239, 2007.

[8] D. A. Vallado and W. D. McClain. *Fundamentals of Astrodynamics and Applications*. Number 21 in Space Technology Library. Microcosm Press, Hawthorne, Calif, 4 edition, 2013. ISBN 978-1-881883-18-0.

[9] L. Pedroso. Distributed decentralized control for very large-scale systems with application to LEO satellite mega-constellations. Master's thesis, Instituto Superior Técnico, University of Lisbon, 2022.

[10] S. Bandyopadhyay, G. P. Subramanian, R. Foust, D. Morgan, S.-J. Chung, and F. Hadaegh. A review of impending small satellite formation flying missions. In *53rd AIAA Aerospace Sciences Meeting*, page 1623. American Institute of Aeronautics and Astronautics, Jan 2015. doi: 10.2514/6.2015-1623.

[11] R. H. Merson. A brief survey of satellite orbit determination. *Mathematical and Physical Sciences*, 262(1124): 71–78, 1967.

[12] NASA. Helioswarm. https://science.nasa.gov/mission/helioswarm, Accessed 2023. NASA's Science Mission Directorate.

[13] S. D'Amico, J.-S. Ardaens, and S. De Florio. Autonomous formation flying based on gps—prisma flight results. *Acta Astronautica*, 82(1):69–79, 2013. doi: 10.1016/j.actaastro.2012.04.033.

[14] C. T. Fraser. *Adaptive Extended Kalman Filtering Strategies for Autonomous Relative Navigation of Formation Flying Spacecraft*. Phd thesis, Carleton University, 2019.

[15] R. Oliveira. Orbit determination for low-altitude satellites using semianalytical satellite theory. Master's thesis, Master's thesis, IST, Universidade de Lisboa, 2021.

[16] G. Battistelli and L. Chisci. Stability of consensus extended kalman filter for distributed state estimation. *Automatica*, 68:169–178, 2016.

[17] J. Wang and E. A. Butcher. Decentralized estimation of spacecraft relative motion using consensus extended kalman filter. In *2018 Space Flight Mechanics Meeting*, page 1965, 2018.

[18] R. Olfati-Saber. Distributed kalman filtering and sensor fusion in sensor networks. In *Networked Embedded Sensing and Control: Workshop NESC'05: University of Notre Dame, USA October 2005 Proceedings*, pages 157–167. Springer, 2006.

[19] R. Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 8179–8184. IEEE, 2005.

[20] S. S. Stanković, M. S. Stankovic, and D. M. Stipanovic. Decentralized parameter estimation by consensus based stochastic approximation. *IEEE Transactions on Automatic Control*, 56(3):531–543, 2010.

[21] D. Dumitriu, S. Marques, P. U. Lima, and B. Udrea. Decentralized, low-communication state estimation and optimal guidance of formation flying spacecraft. In *16th AAS/AIAA Space Flight Mechanics Meetings*, 2006.

[22] K. Matsuka, A. O. Feldman, E. S. Lupu, S.-J. Chung, and F. Y. Hadaegh. Decentralized formation pose estimation for spacecraft swarms. *Advances in Space Research*, 67(11):3527–3545, 2021.

[23] G. Battistelli and L. Chisci. Kullback–leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica*, 50(3):707–718, 2014.

[24] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano. Consensus-based algorithms for distributed filtering. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 794–799. IEEE, 2012.

[25] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano. Consensus-based linear and nonlinear filtering. *IEEE Transactions on Automatic Control*, 60(5):1410–1415, 2014.

[26] R. Olfati-Saber. Kalman-consensus filter: Optimality, stability, and performance. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 7036–7042. Ieee, 2009.

[27] R. A. A. Cordeiro. A low-communication distributed state-estimation framework for satellite formations. Master's thesis, Instituto Superior Técnico, Nov. 2022.

[28] C. G. Lopes and A. H. Sayed. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. *IEEE Transactions on Signal Processing*, 56(7):3122–3136, 2008.

[29] F. S. Cattivelli and A. H. Sayed. Diffusion lms strategies for distributed estimation. *IEEE transactions on signal processing*, 58(3):1035–1048, 2009.

[30] F. S. Cattivelli and A. H. Sayed. Diffusion strategies for distributed kalman filtering and smoothing. *IEEE Transactions on automatic control*, 55(9):2069–2084, 2010.

[31] B. Jia, K. Pham, E. Blasch, G. Chen, and D. Shen. Diffusion-based cooperative space object tracking. *Optical Engineering*, 58(4):041607–041607, 2019.

[32] P. Gil. *Elementos de Mecânica Orbital*. Instituto Superior Técnico, version 1.1.6 (2020-01-11) edition, 2015.

[33] H. D. Curtis. *Orbital Mechanics for Engineering Students*. Elsevier Butterworth-Heinemann, 2005. ISBN 9780750661690.

[34] C. A. Kluever. *Space Flight Dynamics*. John Wiley & Sons Ltd, first edition, 2018. ISBN 9781119157823. Companion website: www.wiley.com/go/Kluever/spaceflightmechanics.

[35] O. Montenbruck and E. Gill. *Satellite Orbits: Models, Methods and Applications*. Springer Berlin Heidelberg, 2011. ISBN 9783540672807.

[36] W. M. Folkner, J. G. Williams, D. H. Boggs, R. S. Park, and P. Kuchynka. The planetary and lunar ephemerides de430 and de431. Tech. Rep. 42-196, NASA Jet Propulsion Laboratory, 2014.

[37] U. Walter. *Astronautics: The Physics of Space Flight*. Springer International Publishing, Cham, 3rd ed., corrected publication edition, 2019. ISBN 978-3-319-74373-2.

[38] R. V. D. Hilst. *Essentials of Geophysics, The Earth's Gravitational Field*. Massachusetts Institute of Technology, Sept. 2004.

[39] V. A. Chobotov. *Orbital Mechanics*, chapter 9, pages 193–213. American Institute of Aeronautics and Astronautics, 3 edition, 2002.

[40] V. A. Chobotov. *Orbital Mechanics*, chapter 8, pages 185–190. American Institute of Aeronautics and Astronautics, 3 edition, 2002.

[41] D. G. Luenberger. *Optimization by vector space methods*. John Wiley & Sons, 1997.

[42] D. Woodbury and J. Junkins. On the consider kalman filter. In *AIAA Guidance, Navigation, and Control Conference*, page 7752, 2010.

[43] R. Zanetti and K. J. DeMars. Joseph formulation of unscented and quadrature filters with application to consider states. *Journal of Guidance, Control, and Dynamics*, 36(6):1860–1864, 2013.

[44] B. Schutz, B. Tapley, and G. H. Born. *Statistical orbit determination*. Elsevier, 2004.

[45] S.-Y. Tu and A. H. Sayed. Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. *IEEE Transactions on Signal Processing*, 60(12):6217–6234, 2012.

[46] F. Paul, B. Jean-Jacques, and S. Lotfi. Various ways to compute the continuous-discrete extended kalman filter. 2012.

[47] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury. Information weighted consensus. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 2732–2737. IEEE, 2012.

[48] L. Xiao, S. Boyd, and S. Lall. Distributed average consensus with time-varying metropolis weights. *Automatica*, 1, 2006.