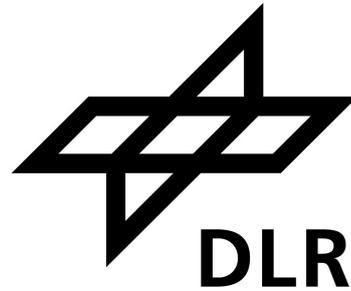


# POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Aerospaziale



**Tesi di Laurea Magistrale**

## **Initial Relative Orbit Determination for Camera-based Navigation in Rendezvous with Uncooperative Target**

**Relatrice:**

Prof.ssa Elisa Capello

**Co-Relatore:**

Dario Ruggiero

**Supervisore Esterno:**

Léo Renaut

**Candidato:**

Lorenzo Porpiglia

**Anno Accademico 2024/2025**

*Per aspera...*

# Abstract

The newest concepts of rendezvous missions involving on-orbit servicing tasks and proximity operations have highlighted the need for alternative strategies for the Target's orbit determination. While a cooperative Target is usually a spacecraft equipped with GNSS sensors and Inertial Measurement Units, and is therefore capable of communicating its state to the Chaser or to the Ground Segment, this does not hold in the case of a non-cooperative Target, like a decommissioned or malfunctioning satellite, an asteroid or a space debris. Angles-only navigation offers a relatively simple and low cost solution to this problem, exploiting a Chaser-mounted optical camera which tracks the Target motion along the entire orbit. The main disadvantage of angles-only navigation is represented by the complex determination of the inter-satellite distance, which is unobservable if linear motion models are employed. This Thesis presents a method for Initial Relative Orbit Determination (IROD) that leverages line-of-sight observations and the non-linearities in the relative motion dynamics to retrieve the initial relative position between the two spacecraft, which can be subsequently used to initialize a real-time navigation filter. Furthermore, given the space hardware requirements and constraints, this method must be efficient and computationally light enough to run on mission-feasible hardware. In the long term, this could enable the design of fully autonomous spacecraft, capable of accomplishing delicate tasks like orbit determination without necessarily relying on data provided *a priori* or uploaded from Ground.

After a brief introduction on rendezvous and proximity operations, with a focus on the contribution of DLR to the history of rendezvous and on-orbit servicing missions, the Thesis will present the theoretical fundamentals of relative motion dynamics in near-circular low Earth orbits, concentrating on analytical relative motion models suited for on-board application. Subsequently, angles-only navigation will be covered thoroughly, with a deeper insight on the Initial Relative Orbit Determination scope, challenges and possible solutions. Chapter 4 will evaluate the accuracy of analytical relative motion models, and will introduce a mathematical formulation which models the nonlinearities associated to the curvature of the orbital path. In Chapter 5, the implemented IROD algorithm will be presented and extensively discussed, highlighting fundamentals, strengths, assumptions and the optimizations that have been performed. Finally, Chapter 6 will offer a detailed evaluation of the algorithm performance, by highlighting its robustness in presence of noise and visibility constraints, presenting tests based on Monte-Carlo analyses, assessing its behaviour with real data from the PRISMA mission of DLR, and finally evaluating its runtime performance on a real onboard computer.



# Contents

<b>Abstract</b>	<b>I</b>
<b>List of Symbols</b>	<b>V</b>
<b>List of Abbreviations</b>	<b>VII</b>
<b>List of Figures</b>	<b>VIII</b>
<b>List of Tables</b>	<b>X</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Space rendezvous . . . . .	1
1.1.1 History of rendezvous missions . . . . .	2
1.2 On-Orbit Servicing . . . . .	3
1.2.1 History of On-Orbit Servicing missions . . . . .	3
1.2.2 Characteristics of On-Orbit Servicing missions . . . . .	4
1.3 Angles-only navigation: the legacy of PRISMA and AVANTI . . . . .	5
1.4 The problem of Initial Relative Orbit Determination . . . . .	6
1.5 Thesis content and outline . . . . .	7
<b>2 Relative Motion Dynamics</b>	<b>9</b>
2.1 Coordinate Frames . . . . .	9
2.1.1 ECI frame . . . . .	9
2.1.2 RTN frame . . . . .	10
2.1.3 Spacecraft’s body frame . . . . .	10
2.1.4 Camera frame . . . . .	10
2.2 Relative motion models . . . . .	11
2.2.1 Hill-Clohessy-Wiltshire Model . . . . .	11
2.2.2 Relative Orbital Elements . . . . .	13
2.2.3 HCW with Relative Orbital Elements . . . . .	16
2.2.4 Alternative motion models . . . . .	16
2.2.5 Relative motion model accounting for $J_2$ . . . . .	18
2.2.6 Relative motion model accounting for $J_2$ and drag . . . . .	18
<b>3 Angles-only Relative Navigation</b>	<b>21</b>
3.1 Angles-only navigation in far-range rendezvous . . . . .	22
3.1.1 Line-Of-Sight measurements . . . . .	22
3.1.2 Critical conditions and limitations . . . . .	24

3.2	The problem of observability . . . . .	25
3.3	State observation and estimation . . . . .	28
3.3.1	Kalman Filter . . . . .	28
3.3.2	Extended Kalman Filter . . . . .	29
3.3.3	Filter initialization . . . . .	30
3.4	Observability for IROD . . . . .	31
3.4.1	Improving the observability . . . . .	31
<b>4</b>	<b>Evaluation of Relative Motion Models</b>	<b>36</b>
4.1	High-Fidelity Simulator . . . . .	36
4.2	HCW model . . . . .	38
4.3	$J_2$ model . . . . .	40
4.4	$J_2$ and drag model . . . . .	42
4.5	Curvilinear coordinate frame . . . . .	45
<b>5</b>	<b>Initial Relative Orbit Determination</b>	<b>52</b>
5.1	Observability enhancement through CCF . . . . .	52
5.2	Algorithm for IROD . . . . .	53
5.2.1	Linear motion model solution . . . . .	54
5.2.2	Non-linear model solution . . . . .	57
5.2.3	Structure of the algorithm . . . . .	59
5.3	Preliminary performance assessment . . . . .	63
<b>6</b>	<b>IROD method performance assessment</b>	<b>67</b>
6.1	Performance with noise . . . . .	67
6.1.1	Sources of noise and errors . . . . .	67
6.1.2	Performance assessment . . . . .	68
6.2	Monte-Carlo analyses . . . . .	71
6.2.1	Analyses set-up . . . . .	71
6.2.2	Assessment of performance in function of measurements parameters . . . . .	72
6.2.3	Assessment of performance in terms of accuracy . . . . .	73
6.2.4	Performance assessment of the linear model solution . . . . .	79
6.2.5	Performance assessment in function of eccentricity . . . . .	80
6.2.6	Performance assessment with respect to noise . . . . .	81
6.2.7	Estimation of initial covariance . . . . .	84
6.3	Evaluation with ARGON data . . . . .	86
6.3.1	Performance with GPS data . . . . .	86
6.3.2	Preliminary performance assessment with ARGON images . . . . .	88
6.4	Runtime evaluation on Zynq Board . . . . .	90
<b>7</b>	<b>Conclusion</b>	<b>93</b>
<b>8</b>	<b>Bibliography</b>	<b>96</b>

# List of Symbols

$\mathbf{A}$	IROD algorithm matrix
$\mathbf{C}$	Matrix mapping ROEs to RTN relative position
$\mathbf{F}$	Force
$G$	Universal gravitation constant
$J$	Loss function
$J_2$	Geopotential second-order zonal coefficient
$M$	Mean anomaly
$\mathbf{M}$	Matrix resulting from the difference between Identity and Projection matrices
$M_B$	Mass of the central body
$N$	Number of Monte-Carlo simulation variants
$\mathbf{P}$	Projection matrix
$\mathbf{P}_0$	Initial covariance matrix
$\mathbf{R}$	Generic rotation matrix
$R_\oplus$	Earth radius
$\mathbf{T}$	Generic transformation matrix
$\mathbf{U}$	LOS vector's cross-product matrix
$a$	Semi-major axis
$\mathbf{b}$	IROD algorithm vector
$\square_c$	Subscript for Chaser
$e$	Eccentricity
$g$	Mapping function from curvilinear to rectilinear coordinates
$\mathbf{h}$	Normalized relative position vector
$i$	Inclination
$m$	Spacecraft mass
$n$	Number of measurements
$\mathbf{r}$	Generic position vector
$\hat{\mathbf{r}}$	Generic position unit vector
$\check{\mathbf{r}}$	Position vector in Curvilinear Coordinate Frame
$t$	Time
$\square_t$	Subscript for Target
$u$	Mean argument of latitude
$\mathbf{u}$	LOS measurement
$\mathbf{v}$	Generic velocity vector
$\mathbf{x}$	Generic state vector

---

$\Delta \mathbf{r}$	Relative position
$\Delta V$	Velocity change due to manoeuvre
$\Phi$	State-Transition Matrix
$\Omega$	Right ascension of the ascending node
$\alpha$	Generic ROEs vector
$\delta \alpha$	Relative orbital elements
$\delta a$	Relative semi-major axis
$\delta \dot{a}$	Relative semi-major axis derivative
$\delta e$	Relative eccentricity
$\delta i$	Relative inclination
$\delta \theta$	Relative ascending node
$\delta \lambda$	Relative mean longitude
$\delta \phi$	Relative perigee
$\epsilon$	Error angle between two vectors
$\theta$	In-plane separation arc-length
$\hat{\theta}$	In-plane angular separation
$\mu$	ROEs generic scaling factor
$\hat{\mu}$	ROEs true scaling factor
$\mu_B$	Gravitational parameter
$\nu$	Mean motion
$\xi$	Signed relative error
$\rho$	Difference between Target and Chaser ECI position
$\sigma$	Standard deviation
$\phi$	Out-of-plane separation arc-length
$\hat{\phi}$	Out-of-plane angular separation
$\psi_x$	Azimuth angle
$\psi_y$	Elevation angle
$\omega$	Argument of perigee

# List of Abbreviations

<b>AOCS</b>	Attitude and Orbit Control System
<b>ARGON</b>	Advanced Rendezvous experiment using GPS and Optical Navigation
<b>AVANTI</b>	Autonomous Vision Approach Navigation and Target Identification
<b>BEESAT-4</b>	Berliner Experimental and Educational Satellite-4
<b>BIROS</b>	Bi-spectral InfraRed Optical System
<b>CCF</b>	Curvilinear Coordinate Frame
<b>CoM</b>	Centre of Mass
<b>DLR</b>	German Aerospace Center
<b>ECI</b>	Earth-Centered Inertial
<b>EKF</b>	Extended Kalman Filter
<b>FOV</b>	Field Of View
<b>GNC</b>	Guidance, Navigation and Control
<b>GNSS</b>	Global Navigation Satellite System
<b>GPS</b>	Global Positioning System
<b>GSOC</b>	German Space Operations Centre
<b>HCW</b>	Hill-Clohessy-Wiltshire
<b>IROD</b>	Initial Relative Orbit Determination
<b>ISS</b>	International Space Station
<b>KF</b>	Kalman Filter
<b>KOEs</b>	Keplerian Orbital Elements
<b>LEO</b>	Low Earth Orbit
<b>LiDAR</b>	Light Detection And Ranging
<b>LOS</b>	Line of Sight
<b>MEV</b>	Mission Extension Vehicle
<b>NASA</b>	National Aeronautics and Space Administration
<b>OOS</b>	On-Orbit Servicing
<b>PIL</b>	Processor-In-the-Loop
<b>PRISMA</b>	Prototype Research Instruments and Space Mission technology Advancement
<b>ROEs</b>	Relative Orbital Elements
<b>RTN</b>	Radial-Tangential-Normal
<b>SNSB</b>	Swedish National Space Board
<b>STM</b>	State-Transition Matrix
<b>SVD</b>	Singular Value Decomposition
<b>TLEs</b>	Two-Line Elements

# List of Figures

1.1	Soyuz TMA-03M docking to the International Space Station (ISS). Credits: NASA . . . . .	1
1.2	Gemini 8 approaching the Agena target vehicle. Credits: NASA . . . . .	2
1.3	Orbital Express: ASTRO and NEXTSat. Credits: Boeing/DARPA . . . . .	3
1.4	Mango (left) and Tango (right). Credits: OHB-SE . . . . .	5
1.5	BIROS (left) and BEESAT-4 (right). Credits: DLR . . . . .	6
2.1	ECI reference frame . . . . .	9
2.2	RTN reference frame . . . . .	10
2.3	Spacecraft's body frame . . . . .	11
2.4	Definition of relative eccentricity and inclination. . . . .	14
2.5	Relative motion with relative eccentricity/inclination vectors. Source: [1] . . . . .	15
3.1	Relative orbit ambiguity in camera based navigation. Source: [1] . . . . .	22
3.2	Visual representation of LOS angles . . . . .	23
3.3	Camera Field of View representation . . . . .	24
3.4	LOS measurement profile ambiguity . . . . .	27
3.5	Different manoeuvres profiles . . . . .	32
3.6	Observability with camera offset . . . . .	33
4.1	Motion of the Target with respect to the Chaser in RTN frame. High Fidelity Simulator . . . . .	37
4.2	Motion of the Target with respect to the Chaser in RN plane. Dynamic Simulator . . . . .	37
4.3	Motion of the Target with respect to the Chaser in RTN frame. HCW model . . . . .	38
4.4	Motion of the Target with respect to the Chaser in RN plane. HCW model . . . . .	39
4.5	Evolution of the ROEs for HCW model . . . . .	39
4.6	Evolution of LOS Error of the HCW model . . . . .	40
4.7	Motion of the Target with respect to the Chaser in RTN frame. $J_2$ model . . . . .	40
4.8	Motion of the Target with respect to the Chaser in RN plane. $J_2$ model . . . . .	41
4.9	Evolution of the ROEs for $J_2$ model . . . . .	41
4.10	Evolution of LOS Error of the $J_2$ model . . . . .	42
4.11	Motion of the Target with respect to the Chaser in RTN frame. $J_2$ and Drag model . . . . .	43

4.12	Motion of the Target with respect to the Chaser in RN plane. $J_2$ and Drag model . . . . .	43
4.13	Evolution of the ROEs for $J_2$ and Drag model . . . . .	44
4.14	Evolution of LOS Error of the $J_2$ and Drag model . . . . .	44
4.15	Evolution of LOS Error of the numerical $J_2$ -and-drag-only propagation	45
4.16	Cartesian and curvilinear coordinates for relative motion. Source: [2]	46
4.17	Evolution of LOS Error of the HCW model in curvilinear frame . . .	48
4.18	Evolution of LOS Error of the $J_2$ and Drag model in curvilinear frame	48
4.19	Comparison between exact and approximate curvilinear transformation	49
4.20	Accuracy difference between the two transformations . . . . .	50
5.1	Residual curves for initial along-track distances of (a) 10 <i>km</i> and (b) 30 <i>km</i> . . . . .	61
6.1	Residual curve with varying noise, 10 <i>km</i> of relative distance . . . . .	69
6.2	Residual curve with varying noise, 30 <i>km</i> of relative distance . . . . .	70
6.3	Relative error of $a\delta\lambda$ in function of the number of LOS measurements acquired . . . . .	72
6.4	Relative error of $a\delta\lambda$ in function of measurements acquisition interval	73
6.5	Distribution of $a\delta\lambda$ signed relative error . . . . .	74
6.6	Distribution of $a\delta\lambda$ relative error . . . . .	74
6.7	Distribution of absolute error on $r_R$ . . . . .	75
6.8	Distribution of absolute error on $r_N$ . . . . .	76
6.9	Durations of LOS acquisition process in terms of orbits and of time .	77
6.10	Accuracy in function of total observation time . . . . .	78
6.11	Accuracy of $a\delta\dot{a}$ determination . . . . .	78
6.12	Relative error on $a\delta\lambda$ with respect to the eccentricity of the Chaser's orbit . . . . .	80
6.13	Error on relative eccentricity with respect to the eccentricity of the Chaser's orbit . . . . .	81
6.14	Noise effect on $a\delta\lambda$ accuracy . . . . .	82
6.15	Noise effect on radial component accuracy . . . . .	83
6.16	Noise effect on normal component accuracy . . . . .	84
6.17	Trajectory of Mango with respect to Tango during ARGON . . . . .	86
6.18	ROEs evolution during ARGON . . . . .	87
6.19	Image from the ARGON experiment. The bright spot in the centre is the Tango spacecraft . . . . .	89

# List of Tables

5.1	Computation time of linear model solution with SVD and least-squares on Intel Core i9-13950HX 2.20 GHz . . . . .	56
5.2	Computation time for IROD on Intel Core i9-13950HX 2.20 GHz . . .	62
5.3	Full IROD and linear model solutions compared with the reference, 10 <i>km</i> separation . . . . .	63
5.4	IROD solution propagated to the final observation time, 10 <i>km</i> initial separation . . . . .	64
5.5	IROD solution compared with the reference, 30 <i>km</i> initial separation	64
5.6	IROD solution propagated to the final observation time, 30 <i>km</i> initial separation . . . . .	64
5.7	IROD solution compared with the reference, 46 <i>km</i> initial separation	65
5.8	IROD solution propagated to the final observation time, 46 <i>km</i> initial separation . . . . .	65
6.1	Maximum LOS error for each simulation. . . . .	68
6.2	Star trackers errors mean and standard deviation. . . . .	68
6.3	GNSS error mean and standard deviation for position and velocity. .	69
6.4	Attitude actuators errors mean and standard deviation . . . . .	69
6.5	IROD results with noise, inter-satellite distance of 10 <i>km</i> . . . . .	70
6.6	IROD results with noise, inter-satellite distance of 30 <i>km</i> . . . . .	71
6.7	Monte Carlo setup parameters . . . . .	71
6.8	Summary of $a\delta\lambda$ relative error in function of initial relative separation	75
6.9	Absolute error on radial component for different initial distance . . .	76
6.10	Absolute error on normal component for different initial distance . . .	77
6.11	Comparison of performance of full IROD method versus linear model solution in terms of ROEs absolute errors . . . . .	79
6.12	Summary of $a\delta\lambda$ relative error in function of initial relative separation, with noise . . . . .	82
6.13	Absolute error on radial component for different initial distance, case with noise . . . . .	83
6.14	Absolute error on normal component for different initial distance, case with noise . . . . .	84
6.15	Standard deviation and variance for the ROEs. . . . .	85
6.16	IROD with ARGON GPS data, along-track distance of approximately 30 <i>km</i> . . . . .	87

---

6.17 IROD with ARGON GPS data, along-track distance of approximately 22 <i>km</i> . . . . .	88
6.18 IROD with ARGON GPS data, along-track distance of approximately 6 <i>km</i> . . . . .	88
6.19 Linear solution performance with LOS from ARGON images . . . . .	89
6.20 Runtime performance on Zynq Board . . . . .	90

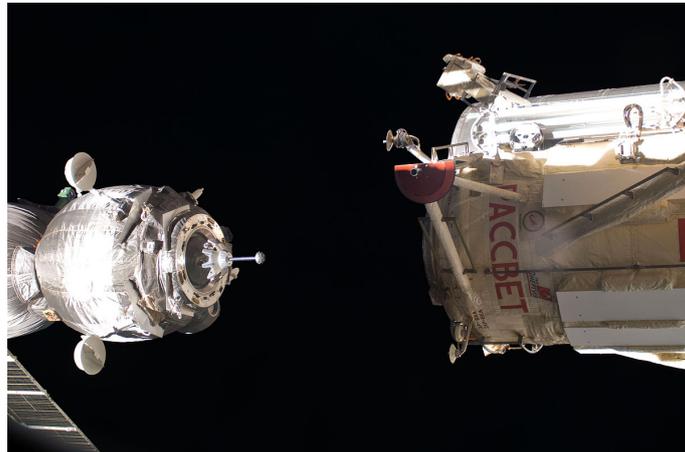


# 1

## Introduction

### 1.1 Space rendezvous

A space rendezvous is a sequence of orbital manoeuvres that enable one spacecraft, commonly known as *Chaser*, to approach another spaceborne object, usually referred to as *Target*. The Target could be a satellite, a space station, an asteroid or a space debris. A rendezvous is essential whenever the Chaser must perform the so-called proximity operations in the vicinity of the Target. Such operations include docking to a designated port (*e.g.* the docking of a *Soyuz* spacecraft to the International Space Station), inspecting the Target with on board dedicated sensors, or capturing the Target with a robotic manipulator in order to perform various functions, such as maintenance, assembly, or component replacement. These latter scenarios fall under the domain of the so-called On-Orbit Servicing (OOS).



**Figure 1.1:** Soyuz TMA-03M docking to the International Space Station (ISS). Credits: NASA

The rendezvous process entails a high complexity, given the strict constraints that must be met. The Chaser spacecraft must be precisely injected by the launch vehicle into an orbit which is coplanar or quasi-coplanar to the Target one. Failure to do so would ultimately result in an unsuccessful mission, since the fuel required to complete the rendezvous at that point would likely exceed the amount stored in the Chaser's tanks. If the rendezvous is successful and the two objects are in

close vicinity to one another, it is crucial to avoid collisions or losses of control of the Chaser: any approach trajectory must be inherently safe, and allow a collision avoidance manoeuvre if required. Furthermore, position and attitude actuators have to be appropriately collocated, calibrated and redunded in order to always allow the completion of delicate and subtle manoeuvres. Finally, the rendezvous sequence must cope with different Sun illumination conditions, communication link constraints and crew work cycle (in the case of a manned rendezvous) [3].

### 1.1.1 History of rendezvous missions

The first attempted rendezvous was made in 1965, when the National Aeronautics and Space Administration (NASA) sent astronaut Jim McDivitt on the *Gemini 4* in an attempt to drive the spacecraft towards its launch vehicle's (*Titan II*) upper stage. The manoeuvre could not be completed, but this failure posed the bases for the first successful rendezvous attempt, when in 1966 astronaut Wally Schirra manoeuvred *Gemini 6* to within less than one meter of its twin spacecraft *Gemini 7*. Few months later, Neil Armstrong successfully docked *Gemini 8* with the *Agena* target vehicle, completing the first rendezvous and docking mission in the history of mankind. One year later, in 1967, the USSR managed to accomplish the first fully automated rendezvous and docking, successfully connecting the *Kosmos-186* and *Kosmos-188* unmanned spacecraft. From that point on, rendezvous missions and experiments grew in number and complexity: from the rendezvous of the Lunar Module with the Command and Service Module during Apollo 11, to the numerous successful docking attempts of the Space Shuttle and of the aforementioned Soyuz spacecraft with the ISS.



**Figure 1.2:** Gemini 8 approaching the Agena target vehicle. Credits: NASA

## 1.2 On-Orbit Servicing

On-Orbit Servicing operations include all those tasks that the Chaser (also referred to as *Servicer*) performs physically and directly on the Target, aimed for example at extending the life, improving the functionality or restore the usability of the Target itself. Such operations include refueling, repairs of hardware or software components, fixing malfunctionalities, installing new components, assembling different pieces together, removing space debris to reduce space junk and de-orbiting decommissioned satellites.

Several studies [4][5][6] have highlighted how relying on OOS could ultimately lead to substantial costs reduction and more efficient resources allocation. For example, on-orbit refueling could extend the life of a certain satellite, preventing the operator from requiring the construction of a new equivalent spacecraft to provide the same exact service. Furthermore, on-orbit repair could allow satellite manufacturers to decrease the redundancy of spacecraft subsystems and components, reducing weight, system complexity and manufacturing costs.



**Figure 1.3:** Orbital Express: ASTRO and NEXTSat. Credits: Boeing/DARPA

### 1.2.1 History of On-Orbit Servicing missions

The first ever satellite servicing mission is often considered to be the the on-orbit repair of the Solar Maximum Mission satellite. In 1984, the NASA Space Shuttle successfully approached the satellite, which had suffered an electronic failure, allowing astronauts James van Hoften and George Nelson to manually repair the malfunctioning spacecraft, ultimately extending the satellite's lifespan by an additional 5 years. In the following years, the famous sequence of manned on-orbit repairs that the Hubble Space Telescope underwent took place, between 1993 and 2009.

In 2007, the United States Defence Advanced Research Project Agency (DARPA) and NASA launched the Orbital Express mission (Figure 1.3). The experiment demonstrated the feasibility of autonomous space servicing: the Boeing-manufactured servicer ASTRO successfully refueled and swapped batteries of the Target NEXTSat.

More recently, Northrop Grumman successfully developed and launched Mission Extension Vehicle (MEV) 1 and 2. In 2018, MEV-1 managed to rendezvous and dock to Intelsat 901, a communication satellite which had been previously collocated in a graveyard orbit, repositioning it to its designated geosynchronous spot and ultimately extending its lifetime by 5 years. Similarly, in 2021 MEV-2 docked to Intelsat 10-02, an actively operating communication satellite, in order to extend its lifespan by approximately 5 years.

### 1.2.2 Characteristics of On-Orbit Servicing missions

OOS missions may differ significantly from one another based on their ultimate goal; however, it is possible to individuate 4 phases which are generally consistent across all. An OOS mission begins with a rendezvous between Chaser and Target, aimed at bringing the former in proximity of the latter. After that, the Chaser starts inspecting the Target to acquire its motion and physical properties information, in order to accurately plan approach, docking or grasping. Following this, the physical contact between the two objects takes place, either with a robotic manipulator or through a docking port. Finally, the Target is stabilized and the servicing operations begin [7]. The complexity of OOS operations varies considerably from case to case, and it is affected by several factors, like environmental factors (orbit altitude, illumination, thermal conditions), servicer capabilities (degree of autonomy, sophistication of the robotic manipulator), and Target design and conditions (presence of docking ports, tumbling motion).

One of the most significant Target conditions to be assessed is the cooperativeness. In the context of the initial rendezvous, approach and inspection, the Target is defined as cooperative if it is fully capable of communicating with the Chaser or with the ground segment, and if it is able to control its own position and attitude. On the contrary, a Target is uncooperative if it is neither able to transmit any information regarding its states, nor capable of exerting authority over its position and attitude. Such scenario is certain if the Target is represented by an asteroid, a space debris or a completely non-functioning satellite. Hybrids between the two cases are also possible: for example, the Target may not be able to control its position and attitude due to a failure of its Attitude and Orbit and Control System (AOCS), but may still be able to communicate. In such case, the Target is still classified as cooperative, but also as uncontrolled.

In the case of an OOS mission with an uncooperative Target, the Chaser's rendezvous Guidance, Navigation and Control (GNC) system has to deal with challenging conditions. Indeed, being the Target just a passive object flying in space, its orbit may be determined approximately with optical or radar tracking from ground, but its precise location has to be determined by the Chaser. The lack of functioning antennas and GNSS sensors on the Target forces the use of optical navigation techniques like angles-only navigation for the determination of its state by the Chaser [8]. Furthermore, the absence of a constant ground station contact requires the GNC system to have a strong level of autonomy, avoiding Man-In-the-Loop interventions as much as possible, ideally only for non-nominal operations.

### 1.3 Angles-only navigation: the legacy of PRISMA and AVANTI

The main features of angles-only optical navigation, often also referred to as Line-Of-Sight (LOS) navigation, will be addressed extensively in Chapter 3. For now, it is sufficient to know that this technique utilizes bearing measurements acquired through a simple optical camera mounted on the Chaser, enabling it to safely navigate towards the Target and complete a full rendezvous manoeuvre. In particular, LOS navigation can be used to approach the Target from several tens of kilometers down to a few hundred meters, covering the so-called far-range and mid-range field. For the close-range field, other specialized sensors shall take over the relative navigation task to determine the Target states [1].

The Prototype Research Instruments and Space Mission technology Advancement (PRISMA) mission was one of the first on-orbit testbeds to demonstrate strategies and technologies for rendezvous and formation flying. The prime contractor was OHB-SE, supported by the Swedish National Space Board (SNSB). Major contributions and support have to be credited to the German Aerospace Center (DLR), the French National Space Center (CNES) and the Danish Technical University (DTU). Several hardware and software experiments involving Mango (the Chaser) and Tango (the Target) were conducted during the mission, involving new technologies for propulsion, vision based sensors (VBS), Global Position System (GPS) and Radio-Frequency (RF) based navigation, as well as several GNC algorithms [9].



**Figure 1.4:** Mango (left) and Tango (right). Credits: OHB-SE

One of the experiments was the Advanced Rendezvous experiment using GPS and Optical Navigation (ARGON), designed and executed by the German Space Operations Center (GSOC) of DLR between the 22nd and the 27th of April 2012. The idea behind ARGON was to demonstrate the feasibility of a LOS navigation-based Ground-In-the-Loop rendezvous with an uncooperative Target: indeed, for the sake of the experiment, Tango was considered passive, and no information from its GPS sensors was directly used during the demonstration. Mango managed to rendezvous with Tango starting from a relative distance of approximately 30 km to a final inter-satellite separation of 3 km, successfully testing the performance of several sensors, of the image processing and of many GNC algorithms and techniques.

The positive outcomes of ARGON posed the bases for the design and realization of the Autonomous Vision Approach Navigation and Target Identification (AVANTI) experiment, conducted in November 2016 by DLR. The AVANTI demonstration was one of the secondary scientific objectives of the FireBird mission, a DLR scientific mission primarily intended for Earth observation. The experiment used the

Bispectral Infrared Optical System (BIROS) spacecraft of the FireBird constellation as Chaser, while the Target was the Berliner Experimental and Educational Satellite-4 (BEESAT-4) cubesat which was carried onboard of BIROS itself. The main difference between ARGON and AVANTI was the near-complete autonomy of the rendezvous between the two spacecraft in the latter experiment. Unlike ARGON, which was carried out with a Ground-In-the-Loop approach, many of the GNC tasks (image processing, Target detection, relative navigation, optimized orbit guidance, safety monitoring) in AVANTI were autonomously conducted on board, with ground activities limited to the choice of the final formation configuration and to some validation and monitoring tasks. Furthermore, while in ARGON the GPS data of Tango, although not used for the rendezvous, were always available in the background and ready to be used if necessary for a collision avoidance manoeuvre, the GPS sensors of BEESAT-4 were completely switched off, making the Target uncooperative in every aspect. During AVANTI, two successful fully autonomous rendezvous were executed: the first brought BIROS from 13 *km* to 1 *km* of distance to BEESAT-4, and the second from 3 *km* to 50 *m*, leveraging angles-only navigation for the approaches [1][10][11].



Figure 1.5: BIROS (left) and BEESAT-4 (right). Credits: DLR

## 1.4 The problem of Initial Relative Orbit Determination

Although in AVANTI the level of autonomy of the whole GNC system reached a very significant level, the navigation algorithms (kinematic Target detection, Extended Kalman Filter) were initialized with an *a priori* estimate of the states and covariance based on Two-Line Elements (TLEs) information. The use of TLEs has two main drawbacks: first, it generally provides low accuracy in determining the relative cross-track and out-of-plane separation between the spacecraft; second, because it has to be provided *a priori* and is not a product of any onboard process, it reduces the overall autonomy of the system [12].

Therefore, it would be beneficial to have a method to derive a coarse estimate of the initial relative state which leverages onboard LOS measurements, allowing the Chaser to autonomously determine the required initialization values of the state vector and of the covariance matrix. This is the main object of the Initial Relative Orbit Determination (IROD) problem, which constitutes the principal topic of this Thesis and of the related research activity.

The relatively recent interest that the problem has arisen implies that very restricted number of studies and solutions have been conducted and proposed so far, with no

in-orbit demonstrations publicly performed and documented. Among the possible approaches explored so far, some exploit complex mathematical frameworks like polynomial dynamics [13] or advanced optimization methods like differential evolution algorithms [14]. On the other hand, other approaches rely on more classic and well-established techniques like batch least-squares refinements [15] or convex optimization methods [16].

## 1.5 Thesis content and outline

This thesis presents the work performed at DLR/GSOC, focused on the implementation and testing of a C++ algorithm for Initial Relative Orbit Determination. The algorithm should be able to operate effectively in real mission scenarios, meaning that it must acquire the necessary LOS measurements in a reasonable amount of time and remain robust against typical noise and errors of angles-only navigation during rendezvous operations. Moreover, considering the limited processing power and storage available in traditional satellite computers, the algorithm should be designed to be computationally efficient, leveraging analytical models and straightforward mathematical techniques.

The Thesis is structured as follows. Chapter 2 will introduce the fundamentals of spacecraft relative motion, covering the different coordinate systems used throughout the work, explaining the Relative Orbital Elements-based relative orbit formulation and presenting several analytical relative motion models that will be utilized for both analysis purposes and algorithm implementation. Chapter 3 will present the main features of angles-only navigation, thoroughly explaining concepts such as LOS measurements and their properties, Kalman Filters and their functioning logic, and finally will address the main challenges in terms of Initial Relative Orbit Determination, focusing on the problem of unobservability and range ambiguity. Chapter 4 will provide a performance assessment of the relative motion models presented in Chapter 2, and will introduce a mathematical formulation capable of increasing their LOS accuracy and, at the same time, of enhancing the observability for IROD. Chapter 5 will discuss extensively the implemented IROD algorithm, highlighting its mathematical backbone, the working principles and the algorithmic optimization which was performed. Finally, Chapter 6 will present a thorough evaluation of the algorithm: the implemented IROD method will be tested in presence of noise, and its global performance will be assessed through an exhaustive Monte-Carlo analysis. Furthermore, it will be tested using data extracted from the PRISMA mission, evaluating its behavior with real data in addition to the simulated ones. Ultimately, the runtime performance will be evaluated by deploying the algorithm on a platform which could be used as satellite on-board computer in the very near future. Chapter 7 will wrap the Thesis up, summarizing key outcomes and addressing future developments and improvements.



## 2

# Relative Motion Dynamics

The purpose of this chapter is to define and describe all the elements necessary to understand the relative motion dynamics of spacecraft, as well as to introduce fundamental concepts that play a crucial role in this Thesis, such as coordinate frames, equations of motion, orbital elements, and perturbations.

## 2.1 Coordinate Frames

### 2.1.1 ECI frame

The Earth Centered Inertial (ECI) frame is the reference frame commonly used to define and describe the absolute dynamics of a spacecraft around the Earth. As the name suggests, it is considered to be an inertial frame fixed with respect to the stars, though it would be more correct to refer to it as *pseudo-inertial* reference frame due to the phenomena of precession of the equinoxes and Earth's nutation. However, since the motion associated with this phenomena is very slow over time, the ECI can effectively be assumed inertial for the applications considered in this Thesis [17].

Naming  $\{\hat{\mathbf{I}}, \hat{\mathbf{J}}, \hat{\mathbf{K}}\}$  the fundamental triad of unit vectors centred at the Earth's Centre of Mass (CoM), the direction of  $\hat{\mathbf{I}}$  is such that, during the vernal equinox, the Earth sees the Sun in the Aries constellation;  $\hat{\mathbf{K}}$  is orthogonal to the fundamental plane, which coincides with the equatorial plane, and points towards the hemisphere containing Polaris;  $\hat{\mathbf{J}}$  completes the right-hand triad. Figure 2.1 presents an illustration of the ECI frame.

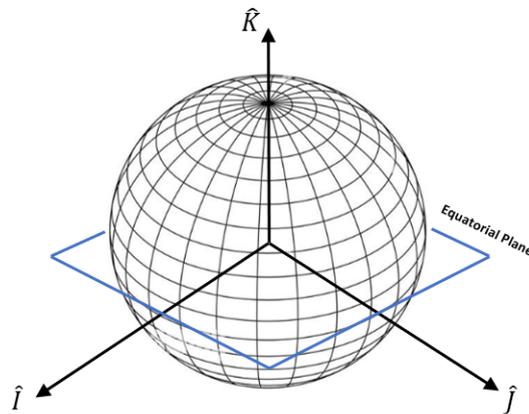


Figure 2.1: ECI reference frame

### 2.1.2 RTN frame

The Radial-Tangential-Normal (RTN) coordinate frame is a local orbital frame used to describe the relative motion of spaceborne objects. By naming  $\{\hat{\mathbf{R}}, \hat{\mathbf{T}}, \hat{\mathbf{N}}\}$  the fundamental triad of unit vectors centred on the CoM of the object (typically a spacecraft),  $\hat{\mathbf{R}}$  is directed as the  $\mathbf{r}$  vector connecting the CoM of the Earth with the CoM of the spacecraft;  $\hat{\mathbf{N}}$  is orthogonal to the fundamental plane (the orbital plane), and is directed as the angular momentum vector  $\mathbf{h}$ ;  $\hat{\mathbf{T}}$  completes the right-hand triad. In the context of this Thesis, the RTN frame is centred on the Chaser spacecraft. The RTN frame can be visualized in Figure 2.2.

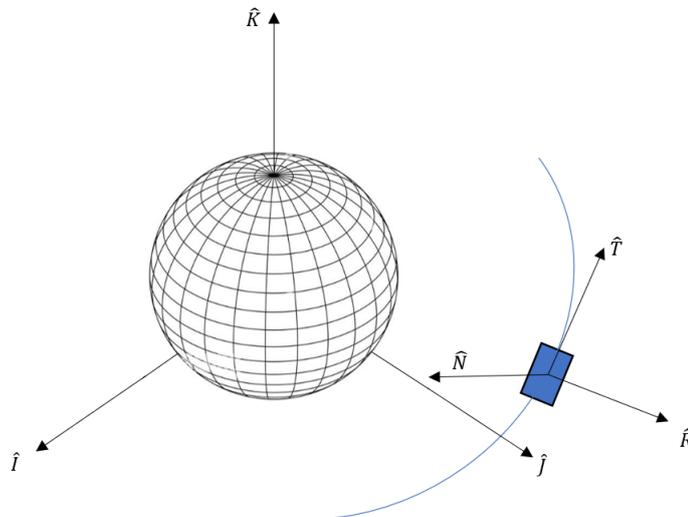


Figure 2.2: RTN reference frame

### 2.1.3 Spacecraft's body frame

The spacecraft body frame is a non-inertial coordinate system used to express the orientation of the object in space. Its fundamental triad  $\{\hat{\mathbf{X}}_B, \hat{\mathbf{Y}}_B, \hat{\mathbf{Z}}_B\}$  is attached to the spacecraft's CoM, with  $\hat{\mathbf{X}}_B$  pointing in the direction in which the sensors are directed,  $\hat{\mathbf{Z}}_B$  such that when  $\hat{\mathbf{X}}_B$  is contained in the orbital plane, so is  $\hat{\mathbf{Z}}_B$ , and  $\hat{\mathbf{Y}}_B$  completes the right-handed orthogonal triad. See Figure 2.3 for a visual representation of the spacecraft's body frame.

### 2.1.4 Camera frame

In the following Chapters, it will become clearer why and how a camera is used as primary sensor for the navigation task. For now, it is sufficient to define the camera reference frame, whose fundamental triad is centred at the centre of the image plane. Being  $\{\hat{\mathbf{X}}_{cam}, \hat{\mathbf{Y}}_{cam}, \hat{\mathbf{Z}}_{cam}\}$  its unit vectors,  $\hat{\mathbf{X}}_{cam}$  and  $\hat{\mathbf{Y}}_{cam}$  are contained in the image plane and point at the top and at the right of the image respectively;  $\hat{\mathbf{Z}}_{cam}$  completes the right-hand triad and points in the direction in which the camera is in boresight.

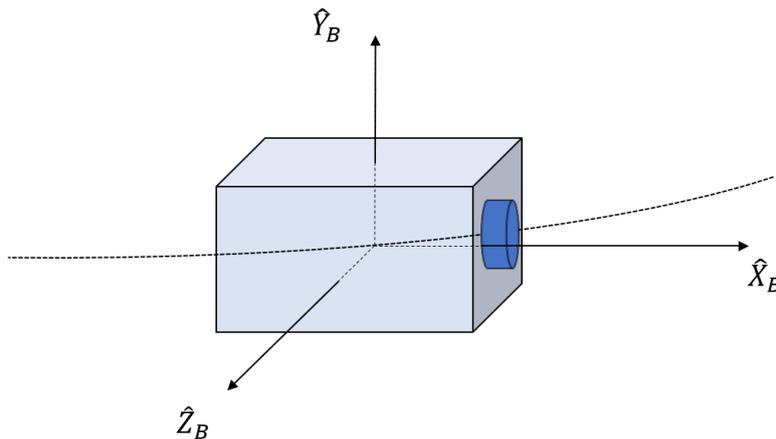


Figure 2.3: Spacecraft's body frame

## 2.2 Relative motion models

When discussing the relative motion of spacecraft, we refer to the movement of one or more spacecraft with respect to another, whose CoM is typically assumed to be the origin of the coordinate frame in which the relative motion is expressed. In case of a formation consisting of two spacecraft, namely a Chaser and a Target, choosing which spacecraft shall be the origin of the reference frame depends mainly on the type of mission that has to be performed. In the context of a rendezvous with a non-cooperative Target, the Chaser spacecraft is commonly the chosen object, since its full state in the inertial frame can be determined with much higher precision than the Target one. Furthermore, the Chaser may be equipped with sensors such as cameras or Light Detection And Ranging (LiDAR) to capture the motion of the Target, making it more reasonable to locate the origin of the frame on the same spacecraft that carries such sensors.

In light of this, from this point forward in this Thesis, an RTN coordinate system centred on the Chaser CoM will be considered as the reference frame in which the relative motion is expressed and analyzed.

### 2.2.1 Hill-Clohessy-Wiltshire Model

It is assumed, from now on, that the ECI state of the Chaser is always known, and expressed as  $\mathbf{x}_c^T = (\mathbf{r}_c^T, \mathbf{v}_c^T)$ , with  $\mathbf{r}_c$  and  $\mathbf{v}_c$  position and velocity vectors in the ECI frame respectively. Analogously, the Target state can be expressed as  $\mathbf{x}_t^T = (\mathbf{r}_t^T, \mathbf{v}_t^T)$ . The motion of an object moving within a central force field and subject to the sole action of the gravitational force of the central body can be expressed by the well-known Newton's Second Law of motion [3]:

$$\ddot{\mathbf{r}} = -\frac{\mu_B}{r^2} \hat{\mathbf{r}}, \quad (2.1)$$

where  $\hat{\mathbf{r}}$  is the position unit vector and  $\mu_B = GM_B$  is the gravitational parameter of the central body, with  $G$  universal gravitation constant and  $M_B$  mass of the central

body. In presence of an external force acting on the spacecraft, naming  $m$  the mass of the spacecraft itself, the equation becomes

$$\ddot{\mathbf{r}} = -\frac{\mu_{\text{grav}}}{r^2}\hat{\mathbf{r}} + \frac{\mathbf{F}_{\text{ext}}}{m}. \quad (2.2)$$

Therefore, denoting with  $\Delta\mathbf{r} = \mathbf{r}_t - \mathbf{r}_c$  the relative position in the inertial frame, the relative motion between Chaser and Target over time can be expressed by

$$\Delta\ddot{\mathbf{r}}(t) = \ddot{\mathbf{r}}_t(t) - \ddot{\mathbf{r}}_c(t). \quad (2.3)$$

However, the solution of this differential equation is not simple, and although an analytical, closed-form solution actually exists [3], it is not very intuitive and not well-suited for every mission. As an example, in the case of a rendezvous mission is more convenient to represent the relative dynamics in a local frame, located either on the Chaser or on the Target.

If one considers the motion as unperturbed, and if assumes that the spacecraft orbits are circular or nearly-circular and if the relative, along-track separation between the two objects does not exceed few tens of kilometers, the motion of the Target with respect to the RTN frame centered on the Chaser can be described by the *Hill-Clohessy-Wiltshire* (HCW) equations [18]:

$$\begin{aligned} \ddot{x} - 2\nu\dot{y} - 3\nu^2x &= 0, \\ \ddot{y} + 2\nu\dot{x} &= 0, \\ \ddot{z} + \nu^2z &= 0, \end{aligned} \quad (2.4)$$

where  $(x, y, z)$  represent the relative position of the Target with respect to the Chaser along the Radial, Tangential and Normal directions respectively, and  $\nu$  represents the Chaser's mean motion. In a more general formulation, naming  $\mathbf{x} = (x, y, z, \dot{x}, \dot{y}, \dot{z})^T$  the state vector, the system can be written as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t). \quad (2.5)$$

This formulation is highly convenient because it enables the calculation of the generic state at time  $t_i$  using only *State-Transition Matrix* (STM) of the system and the state at the initial time  $t_0$ . For a linear time-invariant system, the generic relation that gives the STM  $\Phi$  at the time  $t$  from the state-matrix  $\mathbf{A}$  holds true:

$$\Phi(t, t_0) = e^{\mathbf{A}(t-t_0)}, \quad (2.6)$$

which allows to retrieve the generic state at the time  $t_i$ , from:

$$\mathbf{x}(t_i) = \Phi(t_i, t_0)\mathbf{x}_0, \quad (2.7)$$

where the subscript  $\square_0$  denotes the initial value taken at the reference epoch  $t_0$ . Combining and solving the equations above, calling  $\Delta t = t - t_0$ , it is possible to finally retrieve the State-transition Matrix for the HCW model [19]:

$$\Phi_{HCW}(t, t_0) = \begin{pmatrix} 4 - 3\cos(\nu\Delta t) & 0 & 0 & \frac{1}{\nu}\sin(\nu\Delta t) & \frac{2}{\nu}(1 - \cos(\nu\Delta t)) & 0 \\ 6(\sin(\nu\Delta t) - \nu\Delta t) & 1 & 0 & -\frac{2}{\nu}(1 - \cos(\nu\Delta t)) & \frac{1}{\nu}(4\sin(\nu\Delta t) - 3\nu\Delta t) & 0 \\ 0 & 0 & \cos(\nu\Delta t) & 0 & 0 & \frac{1}{\nu}\sin(\nu\Delta t) \\ 3\nu\sin(\nu\Delta t) & 0 & 0 & \cos(\nu\Delta t) & 2\sin(\nu\Delta t) & 0 \\ 6\nu(\cos(\nu\Delta t) - 1) & 0 & 0 & -2\sin(\nu\Delta t) & 4\cos(\nu\Delta t) - 3 & 0 \\ 0 & 0 & -\nu\sin(\nu\Delta t) & 0 & 0 & \cos(\nu\Delta t) \end{pmatrix}.$$

### 2.2.2 Relative Orbital Elements

It is common practice to describe the motion of a spacecraft around the Earth using the so-called Keplerian Orbital Elements (KOE). These are a set of six parameters that uniquely identify a specific orbit. For a near-circular orbit, these six parameters are:

- semi-major axis  $a$
- eccentricity  $e$
- inclination  $i$
- right ascension of the ascending node  $\Omega$
- argument of perigee  $\omega$
- mean argument of latitude  $u$

The reason that makes this formulation significantly convenient is that the position of the spacecraft over time can be related to quantities describing the size, shape and orientation of the orbit itself, making it more intuitive than a representation in Cartesian coordinates.

Analogously, it would be extremely beneficial to have a similar parametrization for the relative motion: therefore, it is possible to introduce a set of six elements describing the evolution of the relative position between two spaceborne objects. They will be referred to as Relative Orbital Elements (ROEs) [20]. Thus:

- relative semi-major axis  $\delta a$
- relative mean longitude  $\delta \lambda$
- relative inclination  $\delta i$
- relative eccentricity  $\delta e$

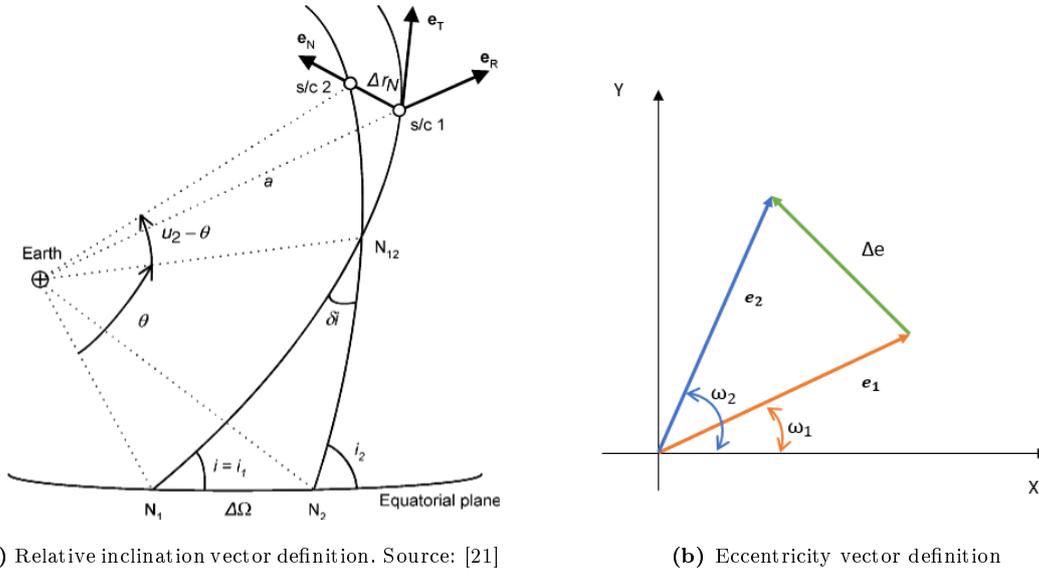
where

$$\delta \mathbf{i} = \begin{pmatrix} i_x \\ i_y \end{pmatrix}, \quad \delta \mathbf{e} = \begin{pmatrix} e_x \\ e_y \end{pmatrix}, \quad (2.8)$$

introducing finally six elements in total. They can be easily derived from the Keplerian Orbital Elements of Chaser and Target, indeed:

$$\delta\boldsymbol{\alpha} = \begin{pmatrix} \delta a \\ \delta\lambda \\ \delta i_x \\ \delta i_y \\ \delta e_x \\ \delta e_y \end{pmatrix} = \begin{pmatrix} (a_t - a_c)/a_c \\ u_t - u_c + (\Omega_t - \Omega_c) \cos i_c \\ i_t - i_c \\ (\Omega_t - \Omega_c) \sin i_c \\ e_{x_t} - e_{x_c} \\ e_{y_t} - e_{y_c} \end{pmatrix} \quad (2.9)$$

where the eccentricity vector is defined as  $\mathbf{e} = (e_x, e_y)^T = (e \cos \omega, e \sin \omega)^T$ .



**Figure 2.4:** Definition of relative eccentricity and inclination.

The geometrical and physical representations of the ROEs are not as simple and straightforward as the ones of the Keplerian orbital elements: nevertheless a brief, concise but exhaustive explanation is provided below.

- The relative semi-major axis  $\delta a$  is the normalized difference between the semi-major axes of the Target and the Chaser.
- The relative mean longitude  $\delta\lambda$  is the angular separation of the two objects on the Chaser's orbit, obtained by projecting the Target's position onto the Chaser's orbital plane.
- the relative inclination can be written as:

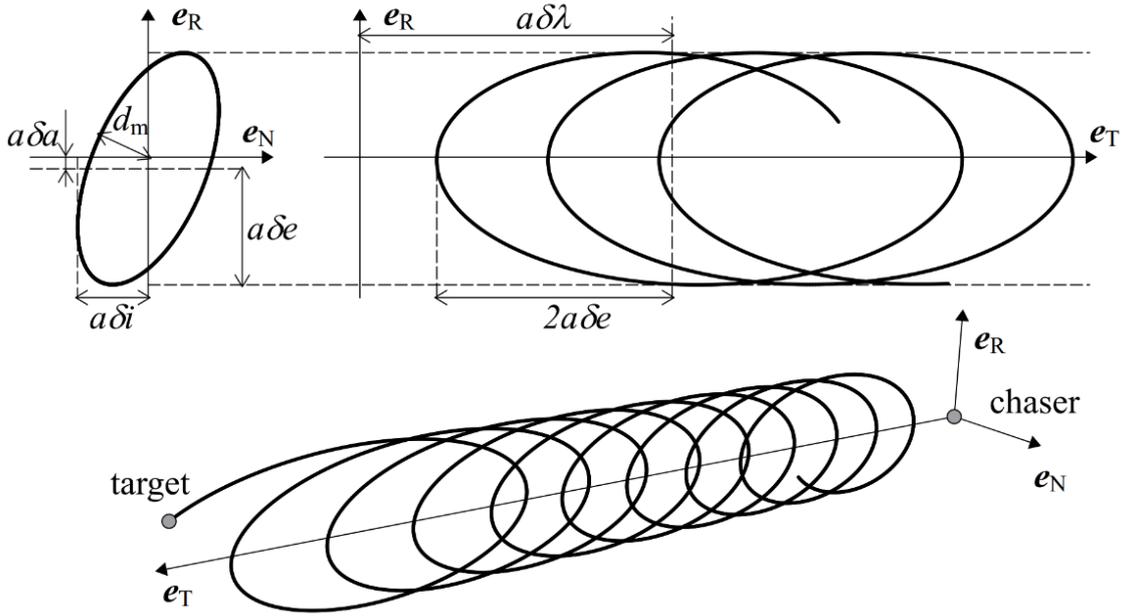
$$\delta\mathbf{i} = \delta i \begin{pmatrix} \cos \delta\theta \\ \sin \delta\theta \end{pmatrix} \quad (2.10)$$

where  $\delta\theta$  is the relative ascending node, defined as the latitude on the Chaser's orbit at which the Chaser crosses the Target's orbital plane.

- the relative eccentricity is the difference between the Target's and Chaser's eccentricity vectors, and can be written as:

$$\delta \mathbf{e} = \delta e \begin{pmatrix} \cos \delta \phi \\ \sin \delta \phi \end{pmatrix} \quad (2.11)$$

where  $\delta \phi$  is called relative perigee: this parameter is defined as the value of the Target's argument of latitude  $u$  for which the Target is located right below the centre of the relative ellipsis in the RT plane, in the case of a relative orbit with  $\delta a = 0$  and  $\delta \lambda = 0$  [20].



**Figure 2.5:** Relative motion with relative eccentricity/inclination vectors. Source: [1]

The link between ROEs and relative Cartesian coordinates, not demonstrated here, can be expressed as follows [20]:

$$\begin{cases} \frac{\delta r_R}{a_c} = \delta a - \delta e \cos(u - \delta \phi) \\ \frac{\delta r_T}{a_c} = \delta \lambda + 2\delta e \sin(u - \delta \phi) \\ \frac{\delta r_N}{a_c} = \delta i \sin(u - \delta \theta) \end{cases} \quad (2.12)$$

The evaluations of the above equations' time derivatives yield:

$$\begin{cases} \frac{\delta v_R}{v_c} = \delta e \sin(u - \delta \phi) \\ \frac{\delta v_T}{v_c} = -\frac{3}{2}\delta a + 2\delta e \cos(u - \delta \phi) \\ \frac{\delta v_N}{v_c} = \delta i \cos(u - \delta \theta) \end{cases} \quad (2.13)$$

### 2.2.3 HCW with Relative Orbital Elements

Previously, it has been highlighted how each ROE is defined by an expression containing one or more Keplerian orbital elements of both Chaser and Target. Defining  $\boldsymbol{\kappa} = (a, e, i, \Omega, \omega, u)^T$ , in absence of perturbations all the orbital elements comprised in  $\boldsymbol{\kappa}$  are constant over time, except for  $u$ :

$$\dot{\boldsymbol{\kappa}} = (0, 0, 0, 0, 0, \dot{u})^T \quad . \quad (2.14)$$

This obviously reflects also on ROEs temporal variation, with only  $\delta\lambda$  varying over time, being it the only ROE dependent from  $u$ :

$$\delta\dot{\boldsymbol{\alpha}} = (0, -\frac{3}{2}\nu\delta a, 0, 0, 0, 0)^T \quad , \quad (2.15)$$

where the formulation of  $\delta\dot{\lambda}$  has been derived considering the approximation  $\dot{u} = \nu$  valid for a near-circular orbit [20]. It can be seen that  $\delta\dot{\boldsymbol{\alpha}}$  exhibits an extremely simple formulation, which can therefore be used to express in a very efficient way the HCW relative motion model. Before doing so, however, it is useful to introduce a slight modification in the formulation of the ROEs vector, which will be used from this point forward. So far, ROEs have been expressed as non-dimensional quantities; however, they can be given a more intuitive form by multiplying the current ROEs state vector  $\delta\boldsymbol{\alpha}$  by the Chaser's semi-major-axis  $a$ :

$$\boldsymbol{\alpha} = a\delta\boldsymbol{\alpha} \quad (2.16)$$

By doing so, it is possible to formulate the HCW motion model in ROEs terms, with the following State-Transition Matrix [22]:

$$\Phi_{HCW}(t, t_0) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{3}{2}\nu\Delta t & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.17)$$

which can be used to retrieve the state vector  $\boldsymbol{\alpha}(t_i)$  at the generic epoch  $t_i$  as:

$$\boldsymbol{\alpha}(t_i) = \Phi(t_i, t_0)\boldsymbol{\alpha}_0 \quad . \quad (2.18)$$

### 2.2.4 Alternative motion models

The strength of the HCW model relies in its simplicity and straight-forwardness, both in its formulation in Cartesian coordinates and with Relative Orbital Elements. However, the assumptions on which it is based are quite strong, and therefore its applicability is limited to some specific cases that include near-circular orbits, small

along-track separation and Keplerian motion. To overcome these limitations, several other models have been developed throughout the past years [23]. Formulations like the one of Yamanaka and Ankersen [24], or the one by Tschauner and Hempel [25] can be used also we dealing with elliptical orbits. On the other side, the adoption of a parametrization which accounts for the ROEs and includes a modeling of the curvature of the orbital path can be used to extend the validity of HCW to inter-satellite distances of several dozens of kilometers. Finally, the assumption of Keplerian motion should be dropped if Low Earth Orbits (LEO) are considered: the LEO environment indeed is characterized by strong perturbing effects, that cause variations of the KOEs over time. The way each element varies over time is unique, but it is possible to find a common ground acknowledging that these variations are composed by a short-period oscillation, a long-period oscillation and a secular variation. If the oscillations are accounted for, then the Keplerian Orbital Elements are labeled as *osculating*; if, instead, accounting for the oscillations is neither necessary nor desired, one could consider only the secular variation, obtaining the so-called *mean* Keplerian Orbital Elements [26].

The most relevant source of perturbation stems from the Earth being non perfectly spherical (*geoid*), which results in a non-homogeneous gravitational field. The common approach for modeling the gravitational field of a planetary body through a representation that considers the zonal harmonics, the tesseral harmonics and the sectoral harmonics that result from the non-homogeneous mass distribution around the centre of the Planet. For the Earth, the first zonal harmonic, named  $J_2$ , has an effect that far exceeds the one of the other remaining terms, therefore it is considered a good approximation to account only for  $J_2$  when assessing this specific disturbance. The KOEs that are most affected by  $J_2$  are the right ascension of the ascending node, the argument of perigee and the mean latitude [27].

Another major contributor among the perturbations is the atmospheric drag: the higher layers of the Earth's atmosphere extend into altitudes considered to be part of the LEO space, with air density progressively decreasing as the distance from the Earth's surface rises. The air is so scattered that it cannot be treated as a continuous fluid; rather, its interaction with the spacecraft has to be modeled at molecular level. Sometimes this determines difficulties in estimating the drag force acting on the spacecraft, although approximate models have been extensively validated and used throughout the years. Qualitatively, it is possible to affirm that the drag acts like a brake on the spacecraft, progressively lowering its semi-major axis and modifying the eccentricity. [28]

There are several other disturbances in LEO, but their effect is relatively negligible compared to  $J_2$  and atmospheric drag. These other perturbations include the third-body effects caused by the Sun's and Moon's gravitational fields, and the solar radiation pressure, caused by the momentum exchange that occurs when photons emitted by the Sun collide with the spacecraft [29][30]. Other disturbances mainly affecting spacecraft attitude (Earth's magnetic field, gravity gradient) are not relevant in the context of this Thesis.

### 2.2.5 Relative motion model accounting for $J_2$

Given the remarkable impact of the  $J_2$  effect on satellites moving in LEO, it would be beneficial to have an analytical relative motion model that incorporates the influence of the Earth's equatorial bulge, mathematically expressed by the  $J_2$  coefficient. From now on, short and long-period oscillations of the ROEs will be neglected, and only mean Relative Orbital Elements will be therefore considered.

Retaining the assumption of near-circular orbit, the relative motion in a  $J_2$  perturbed environment can be expressed with a State-Transition Matrix: recalling that only  $\Omega$ ,  $\omega$  and  $u$  are affected by  $J_2$ , it is indeed possible to obtain a first order relation that links the state vector at  $t_i$  to the initial state vector  $\alpha_0$ :

$$\alpha(t_i) = \Phi_{J_2}(t_i, t_0)\alpha_0. \quad (2.19)$$

The STM  $\Phi_{J_2}$  is structured as follows [31]:

$$\Phi_{J_2}(t, t_0) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ (\zeta + \lambda_A)\Delta t & 1 & \lambda_I\Delta t & 0 & \lambda_E e_{x_0}\Delta t & \lambda_E e_{y_0}\Delta t \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{7}{2}K\Delta t & 0 & 3\gamma\nu\sin^2 i\Delta t & 1 & -\frac{4}{\beta^2}e_{x_0}K\Delta t & -\frac{4}{\beta^2}e_{y_0}K\Delta t \\ -\frac{7}{2}\dot{\omega}A_1\Delta t & 0 & -5KA_1\Delta t & 0 & C + \frac{4}{\beta^2}\dot{\omega}e_{x_0}A_1\Delta t & -S + \frac{4}{\beta^2}\dot{\omega}e_{y_0}A_1\Delta t \\ -\frac{7}{2}\dot{\omega}A_2\Delta t & 0 & -5KA_2\Delta t & 0 & S + \frac{4}{\beta^2}\dot{\omega}e_{x_0}A_2\Delta t & C + \frac{4}{\beta^2}\dot{\omega}e_{y_0}A_2\Delta t \end{pmatrix} \quad (2.20)$$

with

$$\begin{aligned} \gamma &= \frac{J_2 R_\oplus^2}{2a^2\beta^4}, & \beta &= \sqrt{1 - e^2}, \\ C &= \cos(\dot{\omega}\Delta t), & S &= \sin(\dot{\omega}\Delta t), \\ \dot{M}_{J_2} &= \frac{3}{2}\gamma\beta\nu(3\cos^2 i - 1), & \zeta &= -\frac{3}{2}\nu, \\ \lambda_A &= -\frac{7}{2}\frac{\beta + 1}{\beta}\dot{M}_{J_2}, & \lambda_I &= -K(3\beta + 4), \\ \lambda_E &= \frac{3\beta + 4}{\beta^3}\dot{M}_{J_2}, & K &= -\dot{\Omega}\sin i, \\ A_1 &= -e_{x_0}\sin(\dot{\omega}\Delta t) - e_{y_0}\cos(\dot{\omega}\Delta t), \\ A_2 &= e_{x_0}\cos(\dot{\omega}\Delta t) - e_{y_0}\sin(\dot{\omega}\Delta t). \end{aligned}$$

where  $M$  is called *mean anomaly*, and it is given by  $M = u - \omega$ ,  $\dot{M} = \nu + \dot{M}_{J_2}$  and  $R_\oplus$  is the Earth's radius. All the KOEs in the equations above are the Chaser's KOEs at  $t_0$ .

### 2.2.6 Relative motion model accounting for $J_2$ and drag

As explained above, another major effect on satellites' trajectories in LEO is caused by the aerodynamic drag. The drag is often parametrized through a specific parameter called *ballistic coefficient*, which depends on the mass of the object, on its shape,

on the wetted surface and on the flow conditions around the spacecraft. While the ballistic coefficient of the Chaser can be precisely determined due to its well-known mass, geometric features, and finely controlled attitude, this is no longer possible for an uncooperative Target, whose mass and inertia may not be easy to assess and whose attitude is generally uncontrolled. Overall, these factors make it difficult to precisely model the differential drag between the two spacecraft. Nevertheless, if not modeled, differential drag can at least be observed.

As mentioned in Section 2.2.4, drag tends to modify the elements  $a$  and  $e$  of the Keplerian state vector, implying a variation of  $\delta a$ ,  $\delta e_x$  and  $\delta e_y$  when ROEs are considered. However, on a first approximation, it is possible to consider the effect on the eccentricity as negligible, focusing only on  $\delta a$  [10]. To do so, it is convenient to add a seventh ROE which expresses the variation of  $\delta a$ , over time: the state vector becomes:

$$a\delta\boldsymbol{\alpha} = \begin{pmatrix} a\delta\dot{a} \\ a\delta a \\ a\delta\lambda \\ a\delta i_x \\ a\delta i_y \\ a\delta e_x \\ a\delta e_y \end{pmatrix}. \quad (2.21)$$

It is possible to define a STM which accounts for the influence of  $\delta\dot{a}$  on the other 6 ROEs:

$$\boldsymbol{\Phi}_d(t, t_0) = \begin{pmatrix} \Delta t \\ \frac{1}{2}(\zeta + \lambda_A)\Delta t^2 \\ 0 \\ \frac{7}{4}K\Delta t^2 \\ -\frac{7}{4}\dot{\omega}A_1\Delta t^2 \\ -\frac{7}{4}\dot{\omega}A_2\Delta t^2 \end{pmatrix}. \quad (2.22)$$

Combining this STM with  $\boldsymbol{\Phi}_{J_2}$ , it is possible to obtain an analytical non-linear relative motion model which accounts for the main LEO perturbative effects, described by the following State-Transition Matrix [22]:

$$\boldsymbol{\Phi}(t, t_0) = \left( \begin{array}{c|c} 1 & \mathbf{0}_{1,6} \\ \hline \boldsymbol{\Phi}_d(t, t_0) & \boldsymbol{\Phi}_{J_2}(t, t_0) \end{array} \right) \quad (2.23)$$



### 3

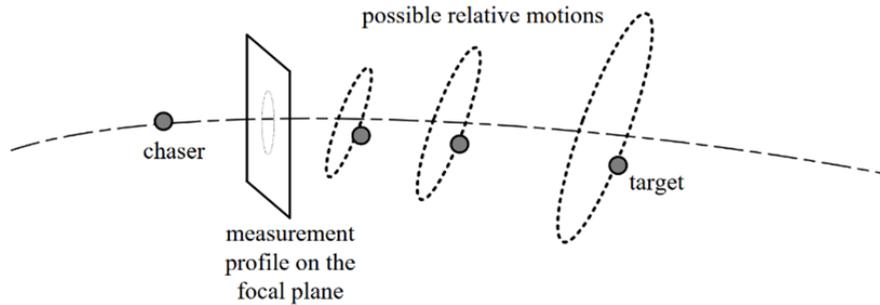
## Angles-only Relative Navigation

Navigation refers to the process of determination of the state of a certain system. In spaceflight, navigation involves the tasks of determining position and attitude of a spaceborne object, achieved by using different kinds of sensors such as Global Navigation Satellite System (GNSS) receivers, star trackers, cameras and LiDARs, exploiting different measurements techniques, accounting for uncertainties and noises. Most satellites are equipped with the so-called Guidance, Navigation and Control (GNC) system, which manages the spacecraft movement, positioning and orientation in space. The definition of Navigation for spacecraft position dynamics has been given previously; for Guidance and Control instead, a short description is provided here for completeness. The role of Guidance is to determine the desired trajectory and to plan and optimize manoeuvres to reach the desired orbit. On the other hand, Control executes specific actions through actuators (usually thrusters) to adjust the trajectory so that it follows the desired one.

Angles-only navigation is a well-established technique used to determine the position of an object with respect to an observer, which has been exploited for thousands of years by humanity. The underlying principle is rather simple: by measuring only the Line-Of-Sight (LOS) angles between a specific reference point and the target object over time, it is possible to determine the relative position and velocity. If the moving object has no capability to control its motion (like in the case of an uncooperative target), and therefore moves obeying to the laws of dynamics, LOS measurements are a powerful tool that can be used to precisely model and predict its trajectory. In case of a rendezvous mission, the instantaneous position of the Target with respect to the Chaser and its evolution over time become the object of the angles-only relative navigation problem [1][32][33].

One of the greatest advantages of this technique when applied to space missions like rendezvous-based OOS missions is the possibility to use a simple, widely commercially available and relatively inexpensive sensor like an optical camera to acquire Target images that can be adequately processed to extract the LOS vector. The main drawback of this method is the possibility that more than one position and trajectory evolution may be associated to the same measurement profile. Indeed, if only LOS angles are measured, the only quantity that can be directly extracted is the direction of the relative position vector between the generic observer and the Target: the magnitude of the vector (in case of a relative position, the distance between the two points) instead, requires additional information to be determined, which

may be not always available. For a rendezvous mission, this issue arises when the 3-dimensional relative orbit of the Target is captured by the camera and projected onto the 2-dimensional image plane, as illustrated in Figure 3.1: the same measurement profile (*i.e.*, the ellipsis on the image plane) corresponds to multiple relative trajectories (*i.e.*, the possible ellipses on which the Target may be moving) [1].



**Figure 3.1:** Relative orbit ambiguity in camera based navigation. Source: [1]

## 3.1 Angles-only navigation in far-range rendezvous

### 3.1.1 Line-Of-Sight measurements

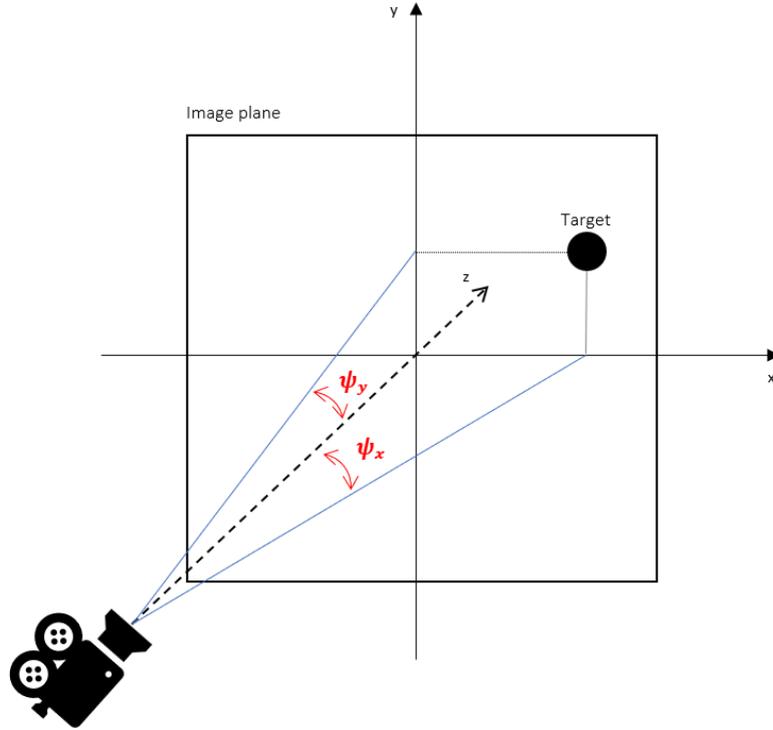
The angles-only navigation task during a rendezvous with an uncooperative Target aims at finding the relative state vector  $\mathbf{x}(t)$  between the Chaser and the Target spacecraft by acquiring a set of  $n$  LOS measurements  $\mathbf{u}_i$  at different epochs  $t_i$  with an optical camera placed on board of the Chaser spacecraft. The more these LOS measurements are close to the actual LOS vector, the more the navigation task can be performed with high accuracy and confidence [12].

Consequently, it is necessary to define the aforementioned actual LOS vector. This is nothing but the relative position vector from the camera optical centre to the Target CoM. The position vector of the camera in the ECI frame is named  $\mathbf{r}_{cam}$ , and  $\mathbf{r}_t$  will be the position vector of the Target's CoM in the same coordinate system. It must be acknowledged that in reality the position of the CoM of an uncooperative object cannot be determined; the sensor is only capable of determining the centroid of the Target image, but not the CoM's object position. Nevertheless, at far range it is possible to approximate the CoM with the Target image centroid, without introducing any observable error. Doing so, it is possible to define the LOS vector at time  $t_i$  as [1]:

$$\mathbf{h}(t_i) = \frac{\mathbf{r}_{cam}(t_i) - \mathbf{r}_t(t_i)}{\|\mathbf{r}_{cam}(t_i) - \mathbf{r}_t(t_i)\|}. \quad (3.1)$$

As stated before, the LOS vector can be computed if two LOS angles are known. These two angles can be referred to with different names depending on the nature of the problem which is being addressed; in the context of this Thesis, they will be called *azimuth* ( $\psi_x$ ) and *elevation* ( $\psi_y$ ). With respect to the Chaser orbital plane,

they represent the in-plane and out-of-plane angular separation respectively. Figure 3.2 provides a visual representation of what has been explained above:



**Figure 3.2:** Visual representation of LOS angles

What has just been defined for the actual LOS vector can be extended to the LOS measurement vector: the camera measures the azimuth and elevation angles of the Target with respect to the optical centre of the image plane, starting from the Target's pixels coordinates on the image. If the Target covers more than one pixel, its centroid's position needs to be estimated first. The angles have to be somehow related to a position unit vector that expresses the Target position with respect to the camera frame; therefore, it is possible to write:

$$\mathbf{y} = \begin{pmatrix} \psi_x \\ \psi_y \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{u_x^{cam}}{u_z^{cam}}\right) \\ \arctan\left(\frac{u_y^{cam}}{u_z^{cam}}\right) \end{pmatrix}. \quad (3.2)$$

Remembering that  $z$  is defined as the boresight direction, it is easy to retrieve the measured unit position vector in the sensor frame, by calculating its norm assuming  $u_z^{cam} = 1$ :

$$\|\mathbf{u}^{cam}\| = \sqrt{\tan^2(\psi_x) + \tan^2(\psi_y) + 1}. \quad (3.3)$$

The unit LOS vector is finally given by:

$$\hat{\mathbf{u}}^{cam} = \frac{(\tan(\psi_x), \tan(\psi_y), 1)^T}{\|\mathbf{u}^{cam}\|} \quad (3.4)$$

It is trivial to say that, in real applications, the measured LOS vector  $\mathbf{u}_i$  will most likely not be precisely matching the actual LOS vector  $\mathbf{h}_i$ , given factors such as sensor noise, pointing errors, or unprecise centroiding. Thus, the quality of the measurements can be assessed by calculating how much the measured position vector deviates from the real position vector. This can be formulated mathematically by computing the angle  $\epsilon$  between the two unit vectors:

$$\epsilon_i = \arcsin (\|\mathbf{u}_i \times \mathbf{h}_i\|). \quad (3.5)$$

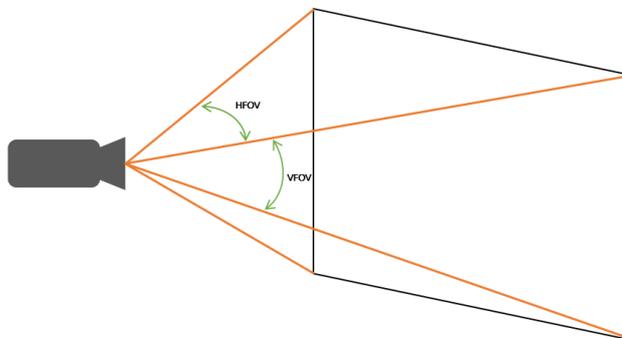
If theoretically one could achieve a perfect measurement, the two vectors would be superimposed and the sine of the angle between them would be zero. It is important to remind that the vectors that have been considered here were already normalized; however, the same relation would hold true if considering the relative position vector  $\mathbf{r}_i$  instead of its normalized counterpart  $\mathbf{h}_i$ .

### 3.1.2 Critical conditions and limitations

When dealing with camera-based navigation, it should be acknowledged that, alongside the numerous advantages provided by this technique, there are several limitations that arise from the operating environment conditions and from the characteristics of the sensor which is being utilized.

First and foremost, the Target has to be visible: in other words, the Target has to be in sunlight in order to perform angles-only navigation with an optical camera. This rather simple but critical aspect poses a fairly big challenge in the application of angles-only navigation in a generic LEO orbit scenario. For instance, a rendezvous in a Sun-synchronous dusk-dawn orbit would encounter minimal visibility challenges, whereas the same operation in an equatorial orbit would be significantly affected by alternating lighting conditions [10].

Camera specifications and performance represent other key factors that can influence angles-only navigation. A standard monocular camera will have a certain Field of View (FOV), which is an angle that describes the angular extent of the scene that the sensor is able to capture. For a camera with a rectangular image plane, an Horizontal FOV (HFOV) and a Vertical FOV (VFOV) are commonly defined, as shown in Figure 3.3.



**Figure 3.3:** Camera Field of View representation

Another relevant factor is represented by the camera resolution. It has been highlighted previously how, in the very likely situation in which the Target comprises more than one pixel, a centroid position estimation is necessary to compute the LOS vector: this process will be briefly described here. The acquired image gets scanned pixel by pixel, looking for a pixel whose brightness exceeds a certain threshold value  $I_2$ . Subsequently, all neighbouring pixels which have a brightness higher than another threshold value  $I_1$  lower than  $I_2$  are marked as belonging to the object of interest, also called cluster. Labelling the position of the  $i$ -th pixel in the image plane as  $\mathbf{p}_i$ , it is possible to retrieve the centroid position by computing the average of the cluster's pixels positions weighted by their brightness [34]:

$$\mathbf{c} = \frac{\sum_{i=1}^n I(\mathbf{p}_i)\mathbf{p}_i}{\sum_{i=1}^n I(\mathbf{p}_i)}. \quad (3.6)$$

It can be deduced how a higher image resolution helps in distinguishing the bright pixels from the dark pixels, lowering proportionally the number of pixels with brightness between  $I_1$  and  $I_2$  with respect to the ones with brightness higher than  $I_2$ , thus increasing the centroid estimation's precision.

Additional operational constraints derive from the presence of both the Sun and the Earth. When the camera points directly towards the Sun, the brightness is so intense that the sensor becomes overwhelmed, making it impossible to see or track other objects in the scene. Therefore, the *Sun Exclusion Angle* is defined, as the angle measured from the boresight axis of the camera to the centre of the Sun. If this angle drops under a certain threshold (*i.e.* the camera is pointing approximately towards the Sun), then the sensor is blinded, and navigation cannot be performed. For some sensors, the Sun Exclusion Angle actually represents a limit value to avoid exceeding not to damage the hardware; for others it simply represents a value under which the camera is blinded and therefore not operative.

Regarding the influence of the Earth, the concept is analogous to what has been explained for the Sun. If the Earth enters the FOV of the camera, the light reflected by its surface has an effect similar to the blinding effect of the Sun, making it impossible to see other flying objects and perform the navigation tasks. Again, some sensors shall avoid pointing towards the Earth not to be damaged by the reflected light and by its thermal radiation, while others are just not capable of seeing the Target anymore, if the planet is in the Field Of View [35][36].

## 3.2 The problem of observability

For a generic dynamic system described in the state-space form, it is possible to write:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= f(\mathbf{x}(t), \mathbf{u}(t), t), \\ \mathbf{y}(t) &= g(\mathbf{x}(t), \mathbf{u}(t), t) \end{aligned} \quad (3.7)$$

where  $\mathbf{x}(t)$  is the state vector,  $\mathbf{u}(t)$  is the control input,  $\mathbf{y}(t)$  is the system output and  $f$  and  $g$  are the functions describing the system dynamics and measurements relationships respectively. A system is defined as observable if its initial state  $\mathbf{x}_0$

can be uniquely determined knowing the system outputs and inputs.

In the context of angles-only relative navigation in space, the observability problem pertains to whether the Target's relative position and velocity, or its ROEs with respect to the Chaser, can be determined through the acquisition of LOS measurements over time. In previous sections, it has been highlighted how angles-only measurements cannot provide precise, if any, information regarding the relative range between the observer and the object of interest. It is intended now to provide a demonstration of what stated above, and to analyze different solutions that may be employed to improve the relative motion observability.

First, it is necessary to introduce the definition of LOS measurement profile. It has been seen previously that with LOS measurement it is intended the sensor reading at a precise time  $t_i$ ; a measurement profile is a continuous function describing all the individual measurements over a certain observation time span [33]. Therefore, the acquisition of azimuth and elevation measurements over time will generate a LOS measurement profile which can be expressed in the camera frame as:

$$\hat{\mathbf{u}}^{cam}(t) = \frac{(\tan \psi_x(t), \tan \psi_y(t), 1)^T}{\|\mathbf{u}^{cam}(t)\|}. \quad (3.8)$$

Before proceeding further, it is important to note that while it is reasonable to express LOS measurement profile in the camera frame, the same is not true for the relative position vector. Instead, the relative position vector is more appropriately expressed in a frame such as the Earth-Centered Inertial (ECI) frame or the Radial-Tangential-Normal (RTN) frame. Given the nature of the problem analyzed, which involves a space rendezvous, the RTN frame is the most suitable choice for the following demonstration. Consequently, it becomes necessary to express  $\hat{\mathbf{u}}^{cam}(t)$  in the RTN frame. To achieve this, let  $\mathbf{R}_{CHA}^{CAM}$  denote the rotation matrix that transforms coordinates from the camera frame to the Chaser's body frame. This matrix remains constant over time if the camera is fixed to the spacecraft. Similarly, let  $\mathbf{R}_{RTN}^{CHA}$  represent the rotation matrix that transforms coordinates from the Chaser's body frame to the RTN frame. Using these matrices, it is possible to write the transformation as:

$$\hat{\mathbf{u}}(t) = \mathbf{R}_{RTN}^{CHA} \mathbf{R}_{CHA}^{CAM} \hat{\mathbf{u}}^{cam}(t) = \mathbf{R}_{RTN}^{CAM} \hat{\mathbf{u}}^{cam}(t), \quad (3.9)$$

which is identical by definition to the normalized relative position vector  $\mathbf{h}(t)$ , if the measurements are considered ideal. For the sake of mathematical consistency, it must be mentioned that in reality one would have to use the transformation matrix  $\mathbf{T}_{CHA}^{CAM}$  instead of the rotation matrix, in order to consider also the position offset between the Chaser's CoM and the Camera optical centre. However, since this offset is typically of the order of tens of centimeters, and the problem considered deals instead with distances of several kilometers, the approximation introduces a negligible error. Let's consider now a linear relative motion model expressed with ROEs, like the ones presented in Chapter 2. Recalling that

$$\mathbf{h}(t) = \frac{\mathbf{r}(t)}{\|\mathbf{r}(t)\|}, \quad (3.10)$$

it is necessary to define a mapping matrix able to transform the Relative Orbital Elements state vector  $\boldsymbol{\alpha}(t)$  into a relative position vector expressed in the RTN frame. In linear approximation, such mapping can be expressed as [37]:

$$\mathbf{C}(t) = \begin{pmatrix} 1 & 0 & 0 & 0 & -\cos u & -\sin u \\ 0 & 1 & 0 & 0 & 2 \sin u & -2 \cos u \\ 0 & 0 & \sin u & -\cos u & 0 & 0 \end{pmatrix}, \quad (3.11)$$

where  $u$  is the Chaser's mean argument of latitude. This yields:

$$\mathbf{r}(t) = \mathbf{C}(t)\boldsymbol{\Phi}(t, t_0)\boldsymbol{\alpha}_0, \quad (3.12)$$

which substituted in 3.10 gives:

$$\mathbf{h}(t) = \frac{\mathbf{C}(t)\boldsymbol{\Phi}(t, t_0)\boldsymbol{\alpha}_0}{\|\mathbf{C}(t)\boldsymbol{\Phi}(t, t_0)\boldsymbol{\alpha}_0\|}. \quad (3.13)$$

It should be noticed how, when introducing an arbitrary scaling factor  $\mu > 0$  such that the new initial state is  $\mu\boldsymbol{\alpha}_0$ , the same relative position unit vector, and therefore the same measurement profile, is obtained:

$$\mathbf{h}(t) = \frac{\mathbf{C}(t)\boldsymbol{\Phi}(t, t_0)\mu\boldsymbol{\alpha}_0}{\|\mathbf{C}(t)\boldsymbol{\Phi}(t, t_0)\mu\boldsymbol{\alpha}_0\|} = \frac{\mu\mathbf{C}(t)\boldsymbol{\Phi}(t, t_0)\boldsymbol{\alpha}_0}{|\mu|\|\mathbf{C}(t)\boldsymbol{\Phi}(t, t_0)\boldsymbol{\alpha}_0\|}. \quad (3.14)$$

For any positive scaling factor  $\mu$ , regardless of how many measurements are taken, it is impossible to distinguish which is the real scaled condition that generates the observed measurement profile, if a linear motion model and a linear output mapping function are employed. This result is known in literature as Woffinden's dilemma [33].

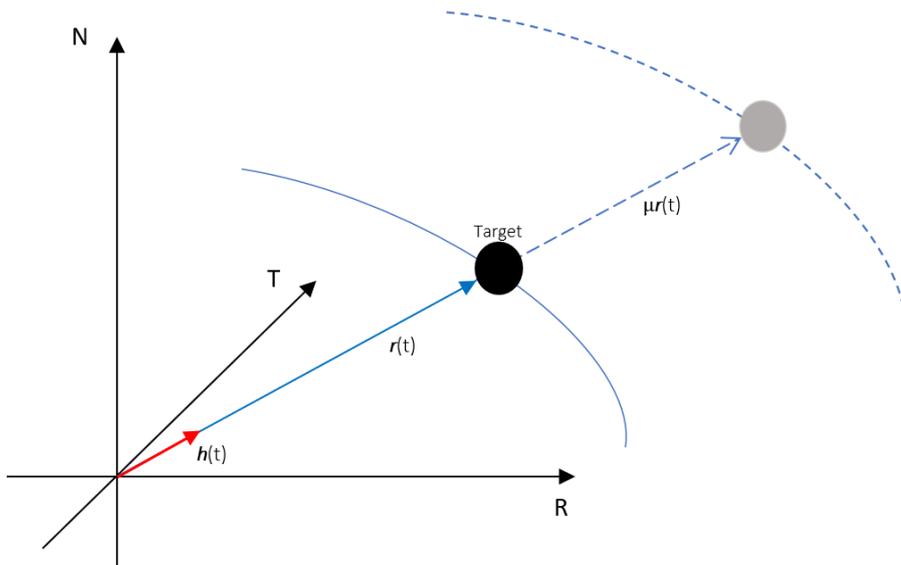


Figure 3.4: LOS measurement profile ambiguity

### 3.3 State observation and estimation

A state observer is essential to effectively perform the navigation task, as it determines the current state of the formation using a set of LOS measurements. While numerous estimation techniques are explored in literature, only a few are well-suited for onboard implementation due to their relatively low computational demands. It is both desirable and practical to use a method capable of estimating the current state in *real time*, a fundamental aspect for autonomous systems. The most famous and employed method is by far the Kalman Filter.

#### 3.3.1 Kalman Filter

Introduced by Rudolf E. Kálmán in 1960 [38], Kalman Filters (KF) are ideal for systems that are continuously evolving. Their biggest advantage is that they require very little memory, since they only need to store information about previous states, making them extremely suited for onboard application, real time issues and embedded systems.

The basic Kalman Filter [39] operates assuming that the system's dynamics is linear, described by a State-Transition Matrix  $\Phi$ : therefore, the evolution of the state vector, in absence of control inputs, can be expressed as:

$$\mathbf{x}_k = \Phi \mathbf{x}_{k-1} + \mathbf{w}_k, \quad (3.15)$$

where  $\mathbf{x}_{k-1}$  represent the state vector at the previous timestep, and  $\mathbf{w}_k$  is the process noise, assumed to be Gaussian. The relationship between the state vector  $\mathbf{x}_k$  and the measurement vector  $\mathbf{y}_k$  can be expressed as:

$$\mathbf{y}_k = \mathbf{H} \mathbf{x}_k + \mathbf{v}_k, \quad (3.16)$$

where  $\mathbf{H}$  is the observation matrix, which linearly maps the state vector  $\mathbf{x}_k$  to the measurement space, and  $\mathbf{v}_k$  is the measurement noise, assumed to be Gaussian as well. The algorithm is structured in two main steps, called *prediction step* and *update step*.

In the prediction step, the state and the error covariance  $\mathbf{P}$  at time  $k$  are predicted, using the linear state-transition model:

$$\mathbf{x}_k^- = \Phi \mathbf{x}_{k-1} \quad (3.17)$$

$$\mathbf{P}_k^- = \Phi \mathbf{P}_{k-1} \Phi^T + \mathbf{Q}, \quad (3.18)$$

where  $\mathbf{Q}$  represents the process noise covariance matrix and it is usually estimated *a priori*.

During the update step, the state and the covariance matrix are updated based on the acquired measurement at time  $k$ . First, the Kalman gain matrix  $\mathbf{K}$  is computed, as:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}, \quad (3.19)$$

where  $\mathbf{R}$  is the measurements noise covariance matrix, which represents quantities such as sensors inaccuracies and measurements noise and it is computed *a priori*. After that, the state estimate and the covariance estimate are updated as:

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{x}_k^-) \quad (3.20)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \quad (3.21)$$

### 3.3.2 Extended Kalman Filter

It should be noted how the KF, in the form that has been presented here, may not be well suited for systems whose dynamics exhibits a non-linear behaviour. Therefore, different versions of the Kalman algorithm have been developed throughout the years, trying to find a way to account for non-linearities with a higher fidelity. One is the Extended Kalman Filter (EKF), whose working principles are analogue to the regular KF, but with the exploitation of non-linear dynamics and measurements models which are subsequently linearized [40].

The dynamics, in absence of control inputs, can be formulated as:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) + \mathbf{w}_k, \quad (3.22)$$

where  $f$  is a non-linear function describing how the states evolve. Analogously, the measurement model can be written as:

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad (3.23)$$

with  $h$  nonlinear function mapping the system state to the measurements space.

The prediction step is similar to the one of the basic KF, with the main difference that, while previously the matrix  $\Phi$  was directly describing the system dynamics, this time it will be actually the result of a linearization of the function  $f$  around the current estimate. In other words,  $\Phi$  is the Jacobian matrix of  $f$ , obtained linearizing  $f$  around  $\mathbf{x}_{k-1}$ :

$$\mathbf{x}_k^- = f(\mathbf{x}_{k-1}) \quad (3.24)$$

$$\Phi_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\mathbf{x}_{k-1}} \quad (3.25)$$

$$\mathbf{P}_k^- = \Phi_k \mathbf{P}_{k-1} \Phi_k^T + \mathbf{Q}. \quad (3.26)$$

Analogously, in the update step,  $\mathbf{H}_k$  represents the Jacobian matrix of the function  $h$  linearized around the predicted state  $\mathbf{x}_k^-$ :

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\mathbf{x}_k^-}, \quad (3.27)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1}, \quad (3.28)$$

$$\mathbf{x}_k = \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{y}_k - h(\mathbf{x}_k^-)), \quad (3.29)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (3.30)$$

The EKF is therefore able to capture some of the effects that derive from the non-linearity of the dynamics and measurements models. While this is certainly an improvement with respect to the basic KF, the linearization introduces computational complexity, and the first-order Taylor expansion may not be sufficiently accurate in case of highly non-linear systems, leading to divergence. To overcome this limitation, other filters like the Unscented Kalman Filter (UKF) have been developed, but they will not be discussed here.

### 3.3.3 Filter initialization

All Kalman Filters require an initialization with an *a priori* estimate of the state vector  $\mathbf{x}_0$  and an *a priori* initial covariance matrix  $\mathbf{P}_0$ . In particular,  $\mathbf{P}_0$  shall reflect the uncertainty on the initial state  $\mathbf{x}_0$ : for instance, if all the components of the initial state vector are known (*i.e.* there is no uncertainty on the initial state),  $\mathbf{P}_0$  will be a null matrix; on the other hand, if the state is not fully known, the initial covariance matrix will have to be tuned accordingly to the level of uncertainty of each single state variable. In real applications, even if the initial state of a system is fully known, there is always a certain degree of uncertainty which shall not be neglected, leading to a  $\mathbf{P}_0$  matrix which is usually non-null.

This said, the problem of the initial state guess or estimation becomes a critical challenge to deal with. Indeed, given the non-linear nature of the relative navigation problem, an initial guess which is too far from the true state would lead to a divergence of the estimation process. Therefore having a robust, reliable initial guess becomes a crucial factor to perform a successful relative navigation task. While in the case of a cooperative Target, is it reasonable to assume that both satellites would be able to provide their position via Global Positioning System (GPS) sensors or other types of GNSS devices, in the scenario of a rendezvous with an uncooperative Target the problem gets more challenging.

The most intuitive and straightforward way to provide the system with a guess of the initial relative state would be the use of Two-Line Elements. TLEs are a peculiar data format used to enumerate the orbital elements of an Earth-orbiting object at a given epoch. Generally, every spaceborne object in LEO with a diameter larger than 10 *cm* is tracked with the TLEs, being it an active satellite, a non-functioning one or a space debris. Obviously, for non-communicating objects, the TLEs give a coarse estimate of the state, since the orbits are propagated with simplified perturbations models. It has been observed how, in the context of a rendezvous mission, the TLEs may provide relatively accurate information on the along track separation, but it lacks precision when it comes to estimating the cross-track and out-of-plane relative orbit, and this may ultimately lead to a divergence of the estimation process [12].

After these considerations, it would be desirable to find a method able to retrieve the initial relative state of the formation by using the same measurement technique employed by the navigation task itself, namely the angles-only LOS measurements. The Initial Relative Orbit Determination problem has indeed attracted considerable attention in the past years [16][41][42], given not only the possibility to compute the initial solution through a well-established, rather simple and low cost technique like angles-only measurements, but also for the improved autonomy of the system

that its usage would provide. The main drawback, as already discussed in the previous section, is the non observability (or better, the *weak* observability, as it will be highlighted in the following section), that the LOS measurements exhibit.

### 3.4 Observability for IROD

With Equation 3.5, it has been highlighted how a perfect LOS measurement satisfies the condition:

$$\mathbf{u}_i \times \mathbf{h}_i = 0. \quad (3.31)$$

Recalling the definition of  $\mathbf{h}_i$ , without loss of generality it is possible to write:

$$\mathbf{u}_i \times \mathbf{r}_i = 0. \quad (3.32)$$

The adoption of a linear relative motion model like the HCW model, as well as the employment of a linear mapping between the states and the relative position vector, would yield to:

$$\mathbf{u}_i \times (\mathbf{C}(t_i)\Phi(t_i, t_0)\mathbf{x}_0) = 0. \quad (3.33)$$

It is clear that, if  $\mathbf{x}_0$  is a solution, then any scaled initial state vector  $\mu\mathbf{x}_0$  is also a solution, leading to an infinity of solutions matching the LOS measurement profile. In order to fit the measurement profile appropriately, it is necessary to adopt approaches able to guarantee the unicity of the initial state associated with that specific LOS profile. These approaches may involve using different propagation models, different mappings between the states and the relative position, or introducing terms not dependent on  $\mathbf{x}_0$ . By doing so, it is theoretically possible to retrieve a unique initial relative state which matches the LOS measurement profile: specifically, this solution is the only one among the others which minimizes a loss function  $J$ , defined as [16]:

$$J = \sum_{i=1}^n \|\mathbf{u}_i \times \mathbf{h}(t_i, \mathbf{x}_0)\|^2. \quad (3.34)$$

#### 3.4.1 Improving the observability

Section 3.2 has highlighted how a linear motion model is not suited for a relative orbit determination, given its intrinsic non-observability which leads to an infinity of possible solutions. Fortunately, there are several techniques which can be used to improve the observability of the system, although not always viable or suited for the phase of far-range rendezvous.

The execution of manoeuvres is the most intuitive, straightforward and effective way to increase the observability of the problem. The linear system would be described by the equation:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (3.35)$$

with  $\mathbf{B}$  control matrix and  $\mathbf{u}$  control input, which in this context represents an acceleration. The solution to this differential equation is:

$$\mathbf{x} = \Phi(t, t_0)\mathbf{x}_0 + \int_{t_0}^t \Phi(\tau, t_0)\mathbf{B}\mathbf{u}(\tau) d\tau. \quad (3.36)$$

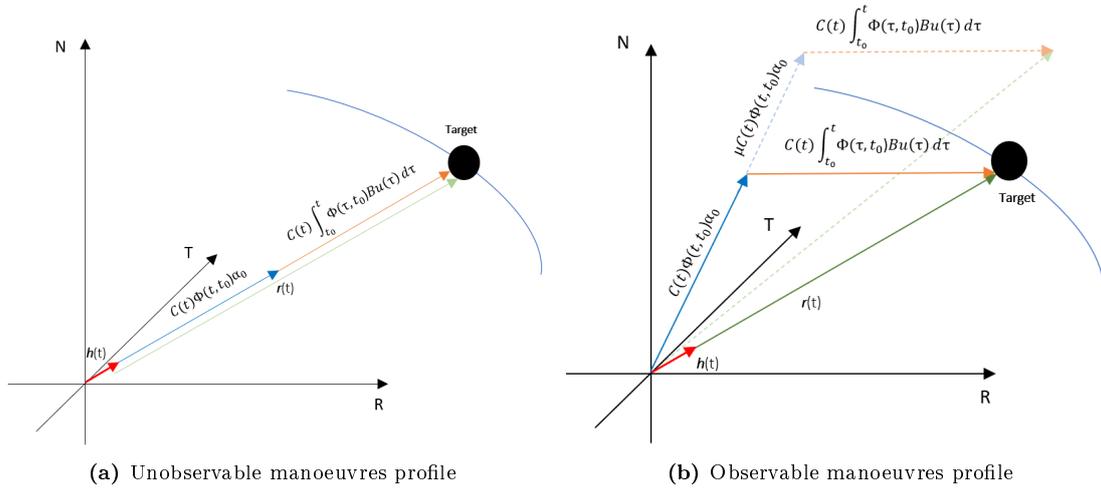
It should be recalled that  $\Phi$  is the STM of a linear motion model. The state vector  $\mathbf{x}$  can also be a set of ROEs  $\boldsymbol{\alpha}$ : this choice will be adopted from now on, in accordance with the rest of this Thesis. Consequently, the LOS unit vector can be expressed as:

$$\mathbf{h}(t) = \frac{\mathbf{C}(t)\Phi(t, t_0)\boldsymbol{\alpha}_0 + \mathbf{C}(t) \int_{t_0}^t \Phi(\tau, t_0)\mathbf{B}\mathbf{u}(\tau) d\tau}{\|\mathbf{C}(t)\Phi(t, t_0)\boldsymbol{\alpha}_0 + \mathbf{C}(t) \int_{t_0}^t \Phi(\tau, t_0)\mathbf{B}\mathbf{u}(\tau) d\tau\|}. \quad (3.37)$$

It should be clear how a generic manoeuvre generates a unique LOS measurement profile, which is therefore associated with to a specific trajectory that could generate only that precise profile over time. If, for instance, a scaled initial condition is selected, the measurement profile in Equation 3.37 is such that:

$$\frac{\mathbf{C}(t)\Phi(t, t_0)\boldsymbol{\alpha}_0 + \mathbf{C}(t) \int_{t_0}^t \Phi(\tau, t_0)\mathbf{B}\mathbf{u}(\tau) d\tau}{\|\mathbf{C}(t)\Phi(t, t_0)\boldsymbol{\alpha}_0 + \mathbf{C}(t) \int_{t_0}^t \Phi(\tau, t_0)\mathbf{B}\mathbf{u}(\tau) d\tau\|} \neq \frac{\mu\mathbf{C}(t)\Phi(t, t_0)\boldsymbol{\alpha}_0 + \mathbf{C}(t) \int_{t_0}^t \Phi(\tau, t_0)\mathbf{B}\mathbf{u}(\tau) d\tau}{\|\mu\mathbf{C}(t)\Phi(t, t_0)\boldsymbol{\alpha}_0 + \mathbf{C}(t) \int_{t_0}^t \Phi(\tau, t_0)\mathbf{B}\mathbf{u}(\tau) d\tau\|}. \quad (3.38)$$

It should be noted that, in reality, not all manoeuvres generate a unique LOS measurement profile: some control inputs may for instance only change the magnitude of the relative position vector  $\mathbf{r}$ , but not its direction, resulting again in a non-observability of the measurement profile. A mathematical formulation which accounts also for these specific cases will not be discussed here, but it can be found in [33]. Figure 3.5 provides a visual representation of what has just been presented.



**Figure 3.5:** Different manoeuvres profiles

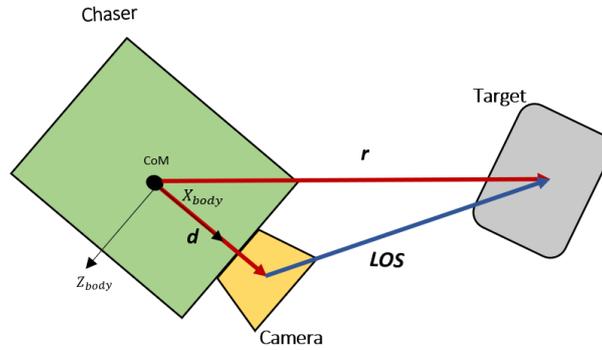
Although the execution of manoeuvres may be a viable approach at first, it should be reminded that this would entail the use of fuel, which at the state of the art cannot be easily replenished once in orbit. Therefore, unless the mission profile allows it, this does not represent a generally feasible solution.

Another possible approach would be leveraging the offset that exists between the

camera and the Chaser's CoM [41]. Without going deep into details, the idea behind this method is that, if the camera and the Centre of Mass are offset by a vector  $\mathbf{d}$  fixed in the Chaser's body frame, it is possible to express the LOS vector in the sensor frame as the sum of  $\mathbf{d}$  and the relative position  $\mathbf{r}$ . Even considering a scaled initial solution,  $\mathbf{d}$  would always remain unchanged, ensuring the unicity of the LOS measurement profile:

$$\mathbf{h}(t) = \frac{\mathbf{C}(t)\Phi(t, t_0)\boldsymbol{\alpha}_0 + \mathbf{R}_{RTN}^{CHA}\mathbf{d}}{\|\mathbf{C}(t)\Phi(t, t_0)\boldsymbol{\alpha}_0 + \mathbf{R}_{RTN}^{CHA}\mathbf{d}\|} \neq \frac{\mu\mathbf{C}(t)\Phi(t, t_0)\boldsymbol{\alpha}_0 + \mathbf{R}_{RTN}^{CHA}\mathbf{d}}{\|\mu\mathbf{C}(t)\Phi(t, t_0)\boldsymbol{\alpha}_0 + \mathbf{R}_{RTN}^{CHA}\mathbf{d}\|}. \quad (3.39)$$

Although rather simple and effective, this method would uniquely work in a close-range scenario, when the spacecraft are separated by a few dozen meters and the order of magnitude of  $\mathbf{d}$  is comparable with the order of magnitude of  $\mathbf{r}$ . In far-range, the measurement profiles accounting or not for  $\mathbf{d}$  would be indistinguishable, leading again to non-observability.



**Figure 3.6:** Observability with camera offset

Since the limitations on the observability of the angles-only navigation come from the fact that the relative motion has been for now considered linear, it is reasonable to assume that a non-linear formulation of the problem would eventually increase the observability. The great advantage of this approach is that it relies solely on the intrinsic characteristics of the relative motion, it only requires an appropriate problem formulation and can be applied to both far-range and proximity operations. However, the observability may not increase as much as it is needed to perform the IROD with sufficient precision and confidence.

It has been illustrated in Chapter 2 how the relative motion models addressed so far do not show a nonlinear dependency on the initial state vector  $\boldsymbol{\alpha}_0$ . Therefore, a possible way around would be to have a motion model non-linearly dependent on the initial condition, so that the STM satisfies  $\Phi(t, t_0, \boldsymbol{\alpha}_0) \neq \Phi(t, t_0, \mu\boldsymbol{\alpha}_0)$  where  $\boldsymbol{\alpha}_0$  is the initial state vector. However, these motion models [23][43] do not account for one big source of non-linearity in the problem considered, which is the curvature of the orbital path [16]. It has been said previously that one of the most important underlying assumptions of the HCW model is that its validity is limited to formations characterized by an along-track separation of few kilometers: this hypothesis has not been relaxed for the other analytical models as well, since it is the consequence of not accounting for the curvature of the trajectory. As a matter of fact, the matrix  $\mathbf{C}(t)$  which maps the ROEs state into a Cartesian relative state is a linear transformation,

which can lead to a substantial degradation of the accuracy of the relative position over time. This suggests that the adoption of a non-linear mapping from ROEs to RTN relative position shall be adopted: this new transformation will be presented in the following Chapter.



# 4

## Evaluation of Relative Motion Models

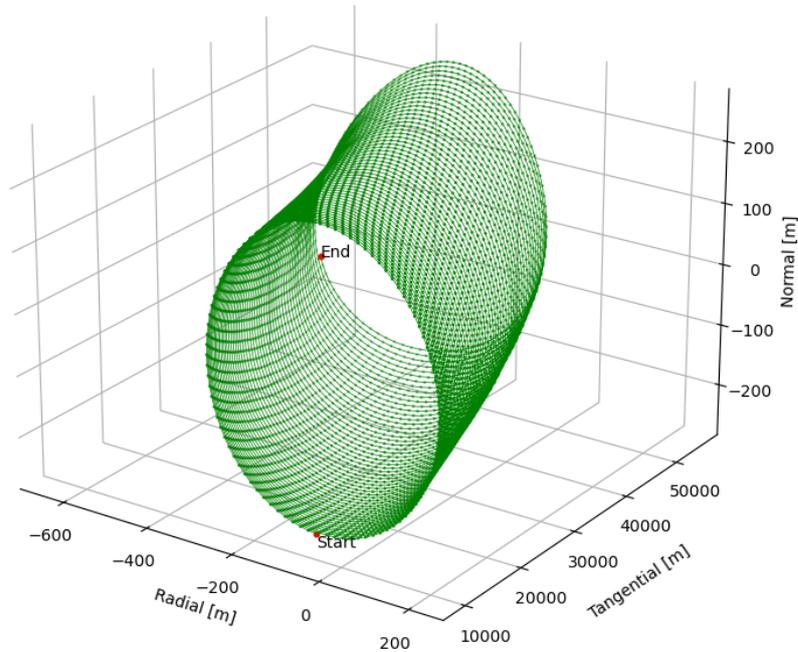
This chapter intends to provide a quantitative evaluation of the accuracy and fidelity of several analytical motion models, like the HCW model or the  $J_2$  and Drag model that have been presented previously. A model of the effect of curvature of the orbital path will be also presented, highlighting the accuracy improvements with respect to the traditional "rectilinear" propagation. The evaluation is performed through a C++ programme which compares the evolution of the states of the analytical relative motion models with a high-fidelity dynamic simulator, which is briefly presented in the following section.

### 4.1 High-Fidelity Simulator

The high-fidelity simulator is a tool developed within DLR/GSOC [44][45]: it is a numerical simulator which uses a Runge-Kutta 4 integrator to propagate the complete state of a satellite (position, attitude and their derivatives), expressed in the ECI frame. The simulator features a precise modeling of the main perturbations which are present in LEO: it accounts for the solar radiation pressure, for the third-body effect due to the presence of the Sun and Moon gravitational fields, evaluates precisely the drag acting on the spacecraft by considering its mass and geometry, accounting also for the atmospheric density variations due to altitude and solar activity, and finally models the Earth non-homogeneous geopotential field with high precision, accounting for a total of 20 subdivisions in the spherical harmonic expansion (degree  $n = 20$ ) and modeling the patterns in the east-west direction up to the order  $m = 20$ . For reference, the Earth's oblateness effect, modeled through the  $J_2$  coefficient, corresponds to the harmonic of degree  $n = 2$  and order  $m = 0$ . The simulator allows also to perform manoeuvres, both for position and attitude, and is able to account for models of several sensors and actuators errors.

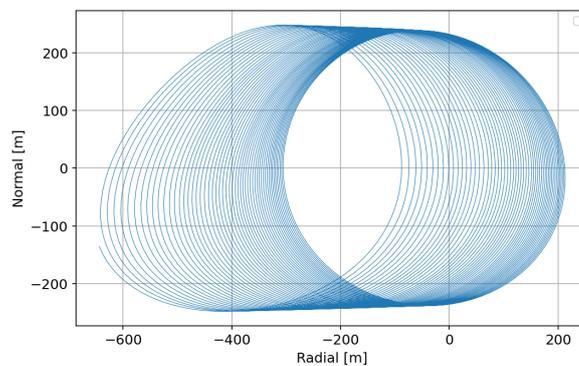
An example of what the simulator can do is provided below. In the considered use-case, which will be used throughout all this Chapter, the two satellites involved in the AVANTI experiment of DLR [46], BIROS (the Chaser) and BEESAT-4 (the Target), are employed. They are placed on two different quasi-circular orbits at an altitude of approximately 500 km, and they are initially separated by an along-track distance of 10 km. Specifically, the initial state vector in terms of ROEs is  $\alpha_0 = (-38.5, 10000, -2.34, 240, -13.3, 260)^T$  m. The negative relative semi-major

axis  $ada$  suggests that the orbit of the Chaser is slightly higher than the orbit of the Target, therefore BIROS is expected to drift away from BEESAT-4 during the propagation, which lasts for approximately 4 and a half days (84 hours). Figure 4.1 shows that, during the considered timespan, the distance between the two objects increases up to 50 *km*. In this use-case, no manoeuvres are performed.



**Figure 4.1:** Motion of the Target with respect to the Chaser in RTN frame. High Fidelity Simulator

The relative cross-track motion can be captured even better when visualizing the relative trajectory in the RN plane, as Figure 4.2 shows. This type of plot is extremely important for the IROD problem, since it represents the part of the Target's motion that can effectively be seen by the camera mounted on the Chaser. In other words, it corresponds to the measurement profile that could be ideally obtained with a perfect, flawless angles-only navigation.



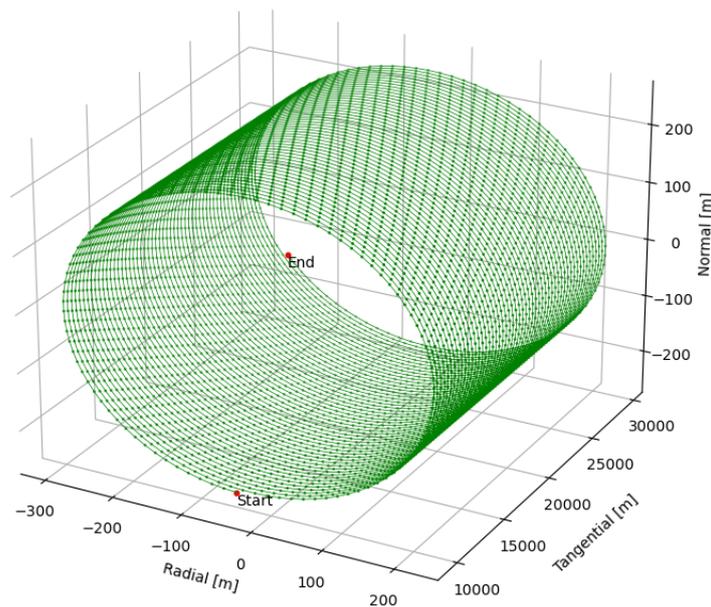
**Figure 4.2:** Motion of the Target with respect to the Chaser in RN plane. Dynamic Simulator

Being a numerical motion model, while exhibiting an extremely high precision in the propagation of the states, it demands a high computational effort to be processed, making it unsuited for an onboard application. On the other hand, while being less precise, the analytical models are computationally light, making them well-suited for spaceborne real time applications.

## 4.2 HCW model

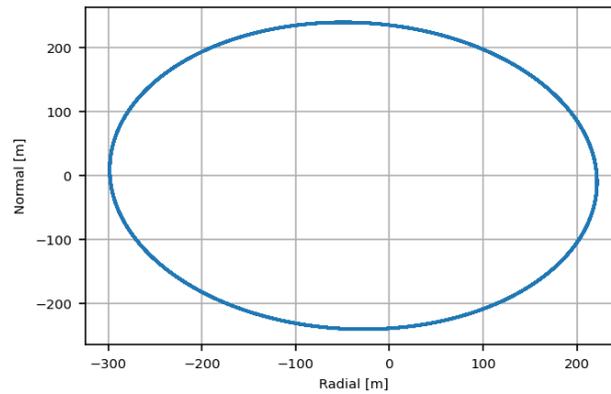
The HCW relative motion model has been already discussed extensively in Chapter 2: it is the most simple model available for the description of the relative motion of two spaceborne objects, and it contains only linear equations. In the HCW model, no perturbations are considered, and the states are updated considering a circular restricted 2-body dynamics, with the only force involved being the Earth's gravitational attraction.

The use-case considered for the propagation is the same utilized above for the high fidelity simulator, and the state vector is updated every  $\Delta t = 1$  s. Figure 4.3 shows that, during the considered timespan, the distance between the two objects increases up to 30 km, which is less than the approximately 50 km obtained with the simulator: this is mainly due to the high differential drag effect featured in this use-case, caused by the low altitude of the orbits and by the massive geometric differences between the satellites. As expected, in absence of perturbations, the formation maintains the same "shape" in the RN plane, with the only time-varying parameter being the along-track separation.



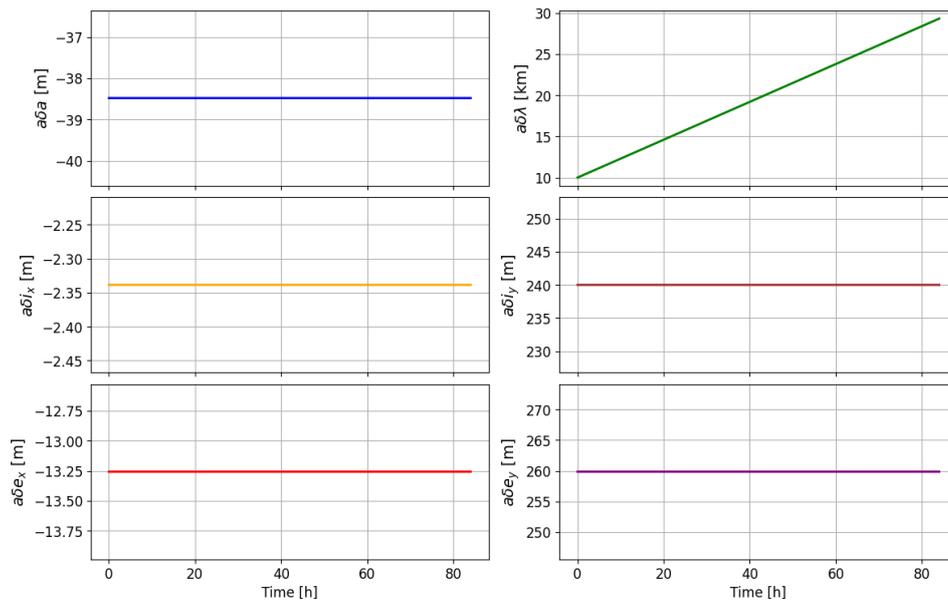
**Figure 4.3:** Motion of the Target with respect to the Chaser in RTN frame. HCW model

The constant shape of the relative cross-track ellipse becomes even clearer when visualizing the trajectory in the RN plane, as Figure 4.4 shows:



**Figure 4.4:** Motion of the Target with respect to the Chaser in RN plane. HCW model

Finally, the evolution of the ROEs is depicted in Figure 4.5: as previously mentioned, the only time-varying parameter is  $a\delta\lambda$ , which grows linearly over time.



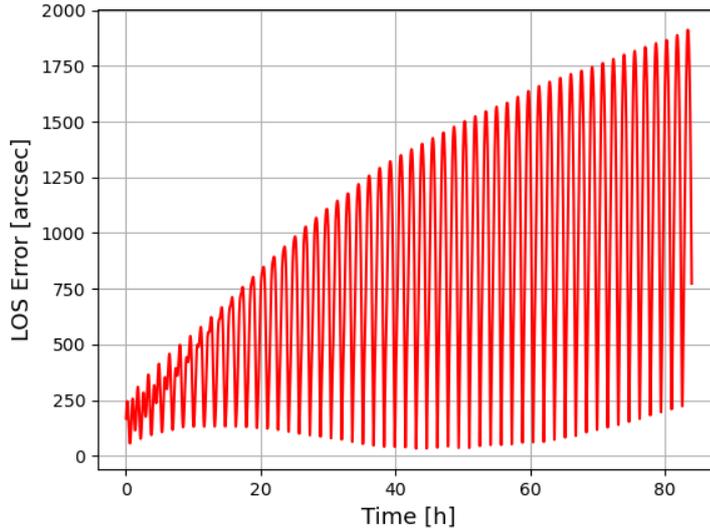
**Figure 4.5:** Evolution of the ROEs for HCW model

For the scope of this Thesis however, the main concern is to assess how much these analytical models deviate from the real evolution of the relative trajectory, here represented by the high-fidelity dynamics simulator: therefore, in order to appropriately test their accuracy, the best way to compare the models with the reference trajectory is to compute the error between the relative position vectors propagated with the analytical models and with the numerical one.

In particular, since angles-only navigation is considered, the key performance parameter to investigate is the LOS error between the two vectors, which can be computed as:

$$\epsilon = \text{atan2}(\|\mathbf{r} \times \mathbf{r}_{ref}\|, \mathbf{r} \cdot \mathbf{r}_{ref}). \quad (4.1)$$

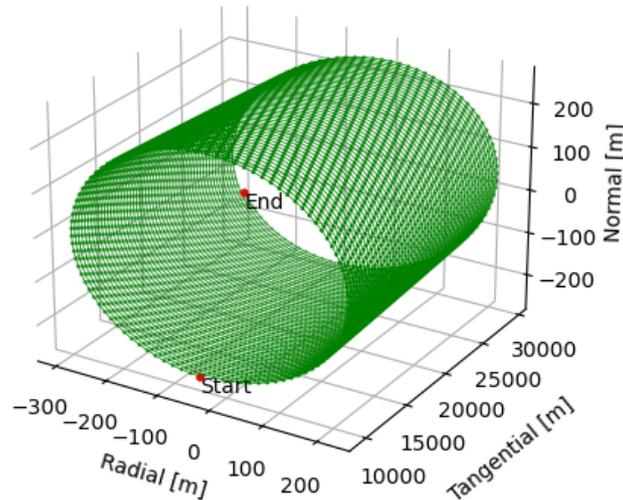
Figure 4.6 shows the LOS error of the HCW model, for the use case considered so far. Even though the evaluation of other models is yet to be presented, it is already clear how HCW is not suited for far range angles-only relative navigation.



**Figure 4.6:** Evolution of LOS Error of the HCW model

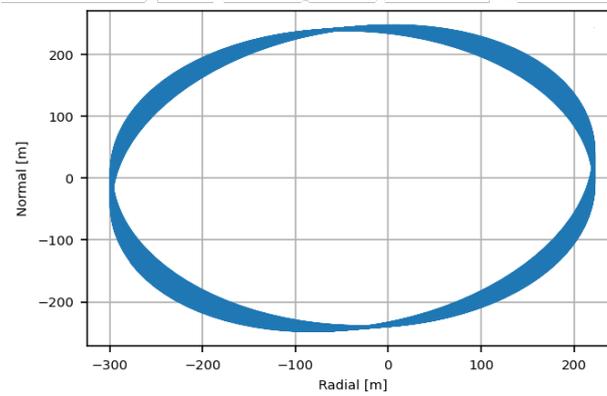
### 4.3 $J_2$ model

The performance evaluation of the  $J_2$  model described in Section 2.2.5 will be now presented. Again, the use case is the same as the one mentioned previously. The evolution in the relative position in the RTN frame at first glance may not look different from the HCW model:



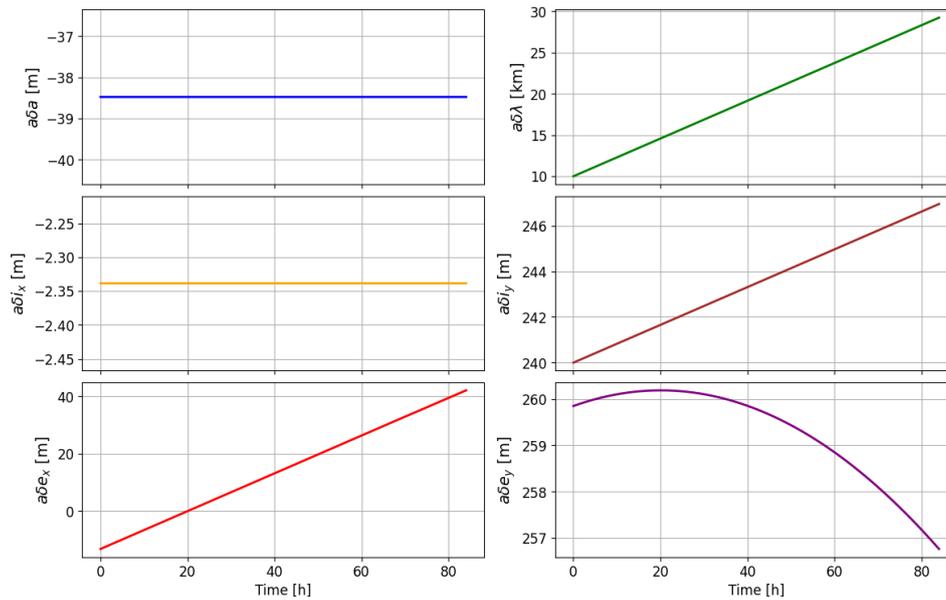
**Figure 4.7:** Motion of the Target with respect to the Chaser in RTN frame.  $J_2$  model

However, the 2D relative motion in the RN plane highlights the differences between the two.



**Figure 4.8:** Motion of the Target with respect to the Chaser in RN plane.  $J_2$  model

As seen in Chapter 2, the analytical formulation of this model is based on the assumption that, due to the  $J_2$  effect, the orbital elements  $\Omega$ ,  $\omega$  and  $u$  are the only one varying over time, while  $a$ ,  $e$  and  $i$  remain constant. Naturally, this reflects on the evolution of the ROEs, that in a  $J_2$ -perturbed environment exhibit the following behaviour:



**Figure 4.9:** Evolution of the ROEs for  $J_2$  model

Overall, the relative trajectory does not differ much from the one obtained with the HCW model, therefore it is reasonable to expect a LOS error similar to the one obtained before: this is indeed the case, as Figure 4.10 shows. Even though the two plots may look identical, the numerical data show minor differences between them, with the  $J_2$  model being slightly more precise than the HCW model. However, the error is still too high to make this model suited for angles-only navigation.

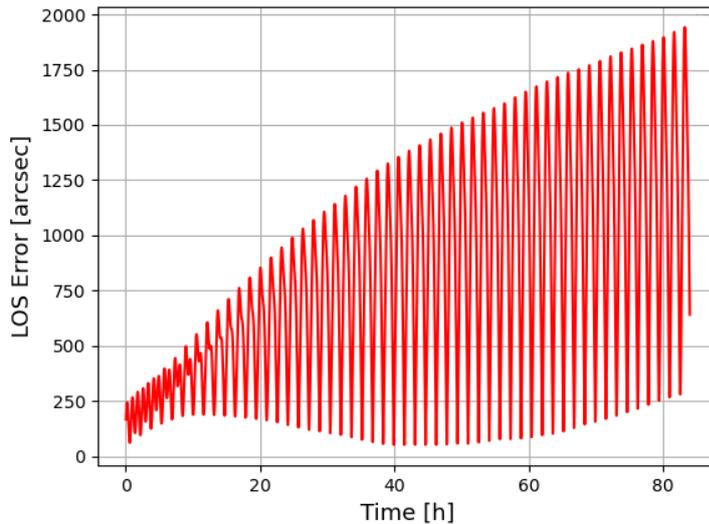


Figure 4.10: Evolution of LOS Error of the  $J_2$  model

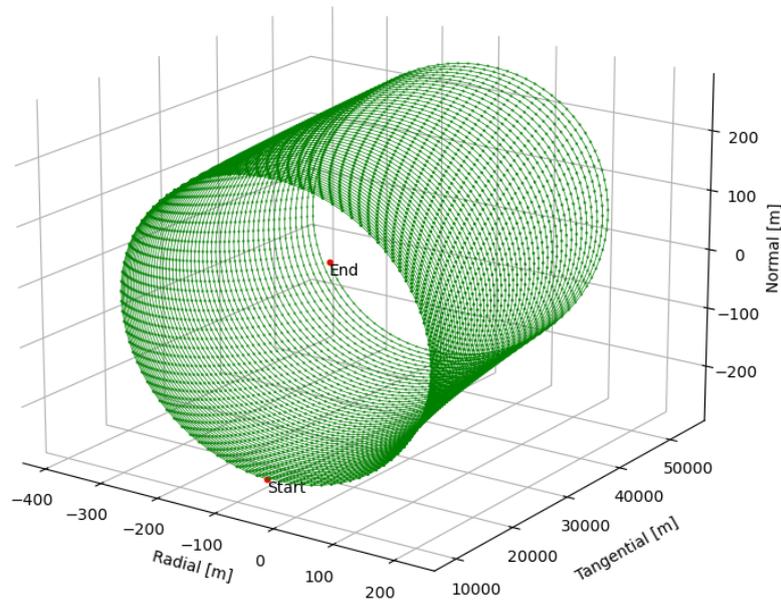
## 4.4 $J_2$ and drag model

It is clear that for the use case considered, neglecting the drag effects leads to extremely unprecise results. Therefore, it is desirable to account also for this perturbing action, especially for those low altitude orbits where the atmospheric density is relatively high. Regardless of the altitude however, including the drag effect would still increase the accuracy of the model, making it a reasonable choice also for higher LEO orbits.

It has been explained in Section 2.2.6 how the differential drag between Target and Chaser can be expressed in terms of ROEs as the temporal derivative of the relative semi-major axis,  $a\delta\dot{a}$ . However, unlike the other ROEs, the differential drag cannot be directly related to the shape of the formation, being it the variation over time of a geometric parameter. Therefore, in order to properly compare the model with the high-fidelity simulator, it is necessary to first retrieve an estimate of the initial differential drag which characterises the considered use-case. To do so, the ECI states of the satellites are propagated with the high fidelity simulator for a sufficiently long timespan  $\Delta t$ , and then the differential drag is computed as the average rate of change of  $a\delta a$  over the considered timespan:

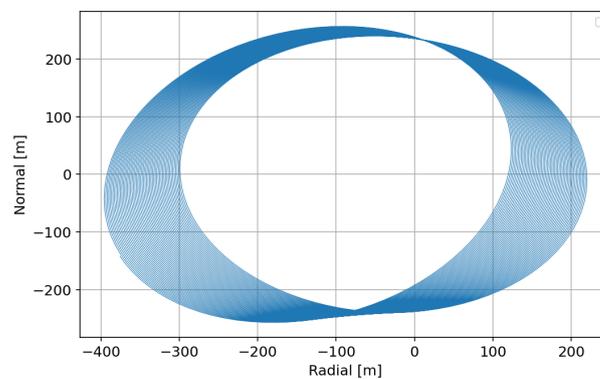
$$a\delta\dot{a} = \frac{(a\delta a)_f - (a\delta a)_0}{\Delta t}. \quad (4.2)$$

It has already been seen how the effect of the drag on the relative eccentricity is not considered in the model. Furthermore, the uncertainties on the attitude of the Target and its time evolution, and the neglect of possible higher order terms make the computed differential drag value an estimation, rather than an accurate result. Nevertheless, the inclusion of this parameter increases the accuracy of the solution, as it will be shown here below. The evolution of the relative trajectory in the RTN frame already shows how this model is actually much more precise than the ones seen previously: indeed, the along track distance here reaches approximately  $50 \text{ km}$  at the end of the simulation, similarly to the high fidelity simulator.



**Figure 4.11:** Motion of the Target with respect to the Chaser in RTN frame.  $J_2$  and Drag model

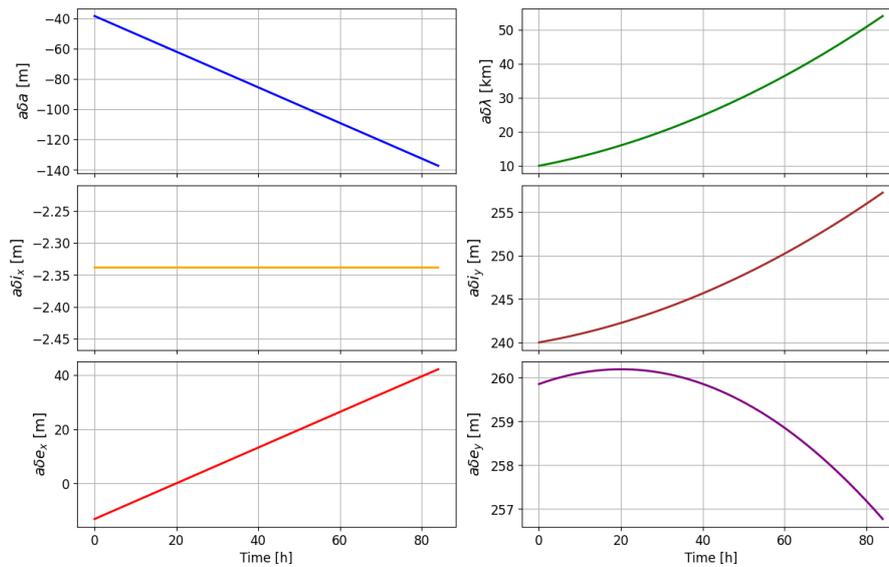
The increased accuracy of this model is even more evident when looking at the trajectory in the RN plane:



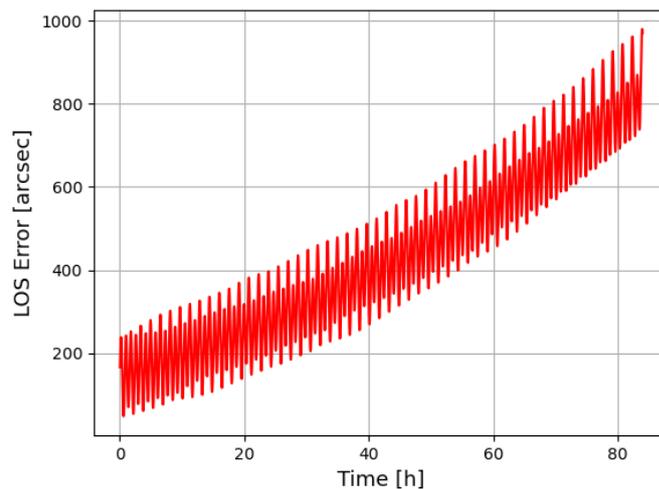
**Figure 4.12:** Motion of the Target with respect to the Chaser in RN plane.  $J_2$  and Drag model

Since the semi-major axis of a satellite is not constant anymore in a drag-perturbed environment, the only constant KOEs are now  $e$  and  $i$ . This implies that all the ROEs will be now time-varying, with the unique exceptions of the differential drag itself, which has been assumed constant, and of  $a\delta i_x$ , which depends solely from the inclination  $i$ .

Finally, a look at the LOS error with respect to the high fidelity simulator (Figure 4.13) definitely shows the improvements introduced by this model. However, the error is still considerably high; the model is not feasible for angles-only navigation in a generic LEO scenario.

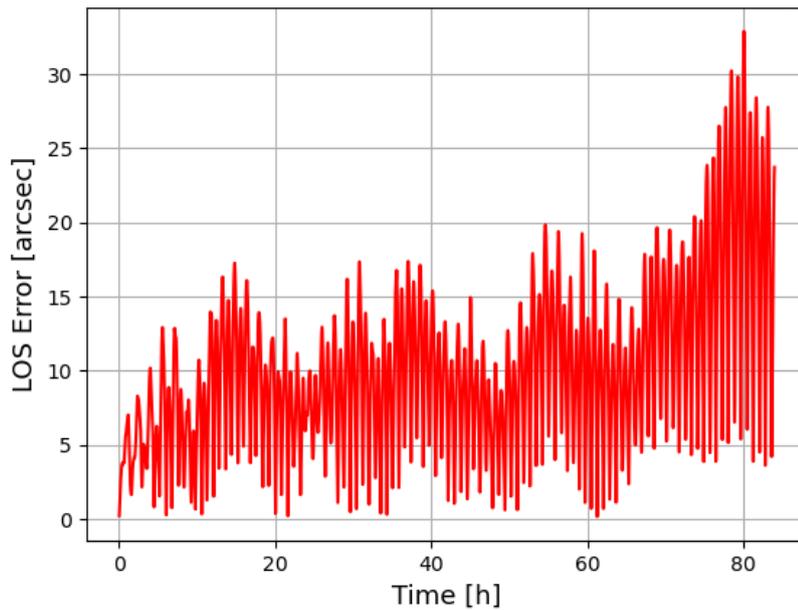


**Figure 4.13:** Evolution of the ROEs for  $J_2$  and Drag model



**Figure 4.14:** Evolution of LOS Error of the  $J_2$  and Drag model

So far, the best analytical relative motion model, which accounts only for the  $J_2$  and Drag disturbances, has been proven to lack the required accuracy to be suited for LOS navigation. The possible reasons behind this may be multiple, and finding the most impacting performance's degrading factor is not trivial. One of the possible candidates is represented by the approximated perturbation model employed, therefore it would be helpful to quantitatively assess the impact of neglecting third body effects, solar radiation and the higher orders terms of the Earth's geopotential field. To achieve this, one can take advantage of the high-fidelity simulator's customizability in defining the involved disturbances and their accuracies. In other words, the idea is to simulate the relative motion with the simulator with only  $J_2$  and drag "switched on", and to compare the result with the trajectory propagated with the whole set of disturbances. The evolution of the LOS Error between the two propagations is reported in the plot in Figure 4.15.



**Figure 4.15:** Evolution of LOS Error of the numerical  $J_2$ -and-drag-only propagation

It is clear that disregarding minor sources of perturbations, while playing an observable effect, cannot be the cause of the poor performance of the analytical models considered so far. With the same set of perturbations, the simulator still outperforms the best model by approximately 30 times. Consequently, it is reasonable to assume at this point that the lack of accuracy of the analytical models has to be due likely to two principal factors: the intrinsic incomparable better performance of numerical models for problems characterized by generally high complexity and non-linearities, and the omission of the consequences of the curvature of the orbital paths. While for the former there is not much that can be done, investigating the latter may lead to performance improvements that could be massive enough to make such models suited for angles-only relative navigation.

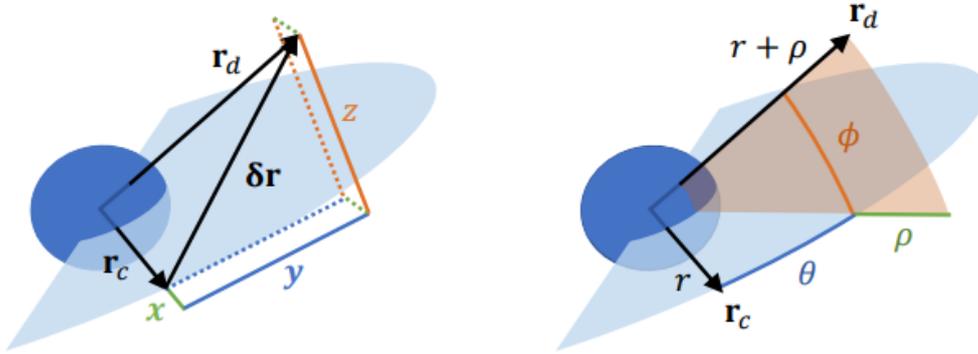
## 4.5 Curvilinear coordinate frame

The scope of introducing a curvilinear coordinate frame (CCF) is double: first, it is auspicious that accounting for the orbits curvature will increase the precision of the analytical propagation models considered so far; second, it has been discussed previously how considering the curvature of the orbital path should make the IROD problem at least slightly observable, while with a classical cartesian frame this is not achievable if a linear relative motion model is considered.

A great advantage of using a curvilinear coordinate frame is that, for circular orbits, the first-order equations of relative motion have the same form that they have in cartesian coordinates; on the contrary, the second order dynamics exhibit substantial differences between the two systems. Nevertheless, this does not concern the models presented so far, as they are derived considering only first-order dynamics.

Multiple curvilinear coordinate frames can be used to meet the needs discussed so far; however, the focus will be on spherical coordinates, illustrated in the image

below.



**Figure 4.16:** Cartesian and curvilinear coordinates for relative motion. Source: [2]

In Figure 4.16,  $\phi$  and  $\theta$  represent the arc lengths of the out of plane and along-track separation respectively. The correspondent subtended angles can be easily retrieved as:

$$\begin{aligned}\hat{\theta} &= \frac{\theta}{r} \\ \hat{\phi} &= \frac{\phi}{r}\end{aligned}\tag{4.3}$$

An exact mapping from the CCF to the RTN coordinate frame considered so far exists [2], and it is given by:

$$\begin{aligned}x &= (r + \rho) \cos \hat{\phi} \cos \hat{\theta} - r \\ y &= (r + \rho) \cos \hat{\phi} \sin \hat{\theta} \\ z &= (r + \rho) \sin \hat{\phi}\end{aligned}\tag{4.4}$$

The opposite transformation, from RTN to CCF, is instead given by:

$$\begin{aligned}\rho &= \sqrt{(r + x)^2 + y^2 + z^2} - r \\ \hat{\theta} &= \arctan\left(\frac{y}{r + x}\right) \\ \hat{\phi} &= \arcsin\left(\frac{z}{r + \rho}\right)\end{aligned}\tag{4.5}$$

While accounting for the exact transformation is fundamental when dealing with large cross-track and out-of-plane separations, for shorter relative distances one may consider an approximated mapping from CCF to RTN: this simple mapping results in the curvature of the orbital path having influence only on the cross-track direction  $x$ , while  $y$  and  $z$  remain unchanged in the two systems ( $y = \theta, z = \phi$ ). This is done by assuming that  $r$  is large compared to  $\rho$ ,  $\theta$  and  $\phi$  so that  $\hat{\theta} \ll 1$  and  $\hat{\phi} \ll 1$ . For  $x$ , is it possible to write:

$$x = (r + \rho) \cos \hat{\phi} \cos \hat{\theta} - r.$$

It is intended now to study what are the effects of having  $\theta$  several orders of magnitude bigger than  $\rho$  and  $\phi$ . Expanding  $\theta$  in Taylor series truncated at the second

order,  $\cos(t) \approx 1 - \frac{t^2}{2}$  for small  $t$ , yields:

$$\cos \hat{\theta} \approx 1 - \frac{\theta^2}{2r^2}.$$

Substituting these into  $x$ :

$$x \approx (r + \rho) \cos \hat{\phi} \left( 1 - \frac{\theta^2}{2r^2} \right) - r.$$

This simplifies to:

$$x \approx (r + \rho) - \frac{(r + \rho)}{2r^2} \theta^2 - r.$$

Since  $\frac{\rho}{r} \ll 1$  for  $r \gg \rho$ , it is possible simplify further, leading to the final approximation of  $x$ :

$$x \approx \rho - \frac{\theta^2}{2r}.$$

For  $y$ , consider:

$$y = (r + \rho) \cos \hat{\phi} \sin \hat{\theta}.$$

Using the small-angle approximation  $\cos(t) \approx 1$  for  $t \ll 1$ :

$$y \approx (r + \rho) \sin \hat{\theta}.$$

Expanding  $\theta$  to the second order results in  $\sin(t) \approx t$  for small  $t$ :

$$y \approx (r + \rho) \hat{\theta}.$$

Since  $r \gg \rho$ , the expression finally yields:

$$y \approx \theta. \tag{4.6}$$

Finally for  $z$ ,

$$z = (r + \rho) \sin \hat{\phi}.$$

Using the small-angle approximation:

$$z \approx (r + \rho) \hat{\phi}.$$

Since  $r \gg \rho$ :

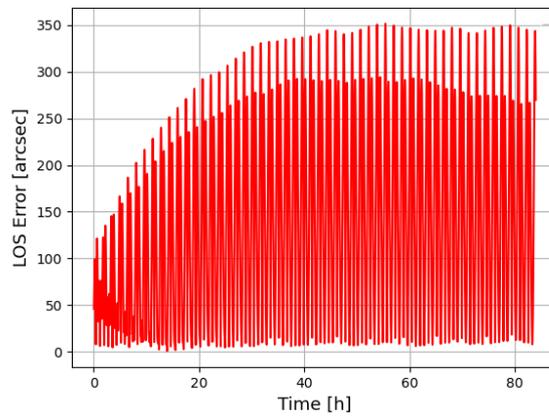
$$z \approx \phi. \tag{4.7}$$

Thus, the approximate transformation from the CCF to the RTN frame can be written as [16]:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \approx \check{\mathbf{r}} + \begin{pmatrix} -\frac{\theta^2}{2R} \\ 0 \\ 0 \end{pmatrix}, \tag{4.8}$$

where  $\check{\mathbf{r}}$  is the relative position vector in the curvilinear coordinates  $(\rho, \theta, \phi)^T$ . As said previously, the scope of introducing the CCF is both to increase the precision and to introduce nonlinearities to make the IROD problem more observable: while the latter aspect will be discussed extensively in the next Chapter, the first can be assessed by propagating the relative trajectories with the motion models presented so far, accounting for the curvature correction given by the CCF. The way the CCF is included in the propagation is defined as follows: since the equations of motions are the same for the Cartesian and for the curvilinear coordinates, the relative trajectory is propagated in the curvilinear system using the same STM. Then, the relative position in curvilinear coordinates is converted in a Cartesian set of coordinates using the aforementioned transformation. In the following digression, the exact mapping from CCF to RTN will be used.

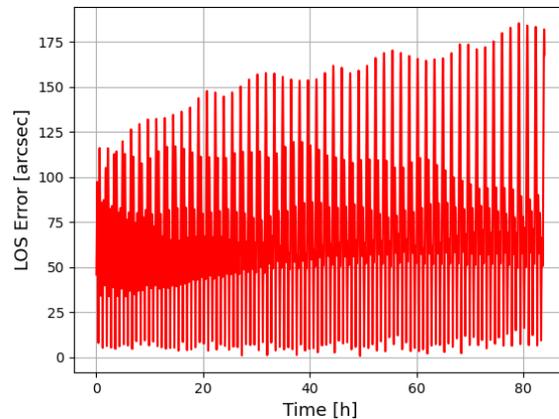
For the HCW model, the LOS error with respect to the high fidelity simulator evolves as shown in Figure 4.17.



**Figure 4.17:** Evolution of LOS Error of the HCW model in curvilinear frame

It can be seen how even a simple motion model like HCW already exhibits more precision than a model accounting for perturbations, if propagated in a frame that better represents the problem's geometries like the CCF.

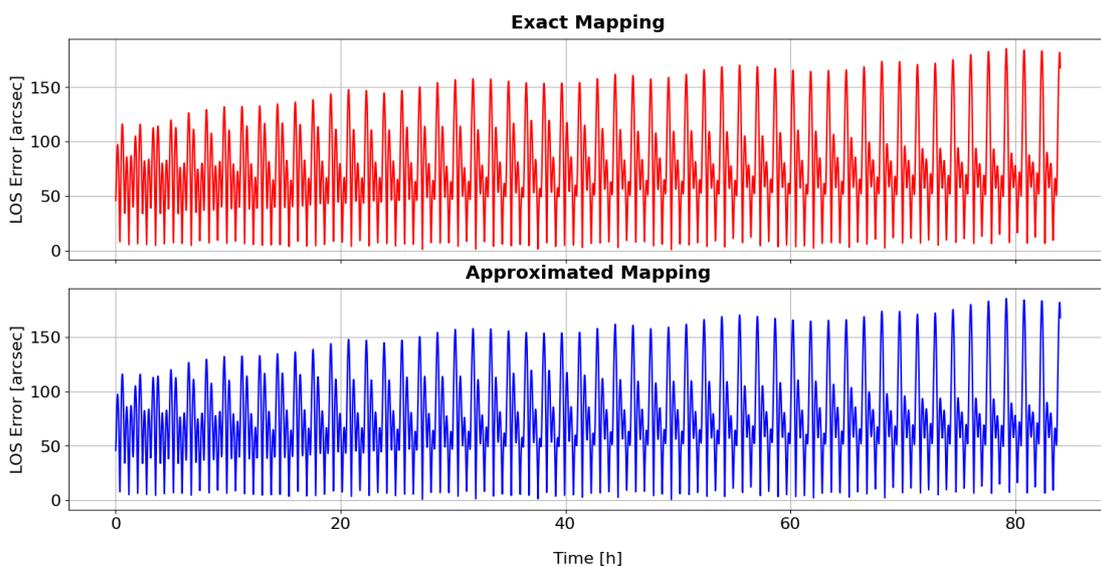
Analogously, the LOS error evolution for the  $J_2$  and Drag model shows the following behaviour:



**Figure 4.18:** Evolution of LOS Error of the  $J_2$  and Drag model in curvilinear frame

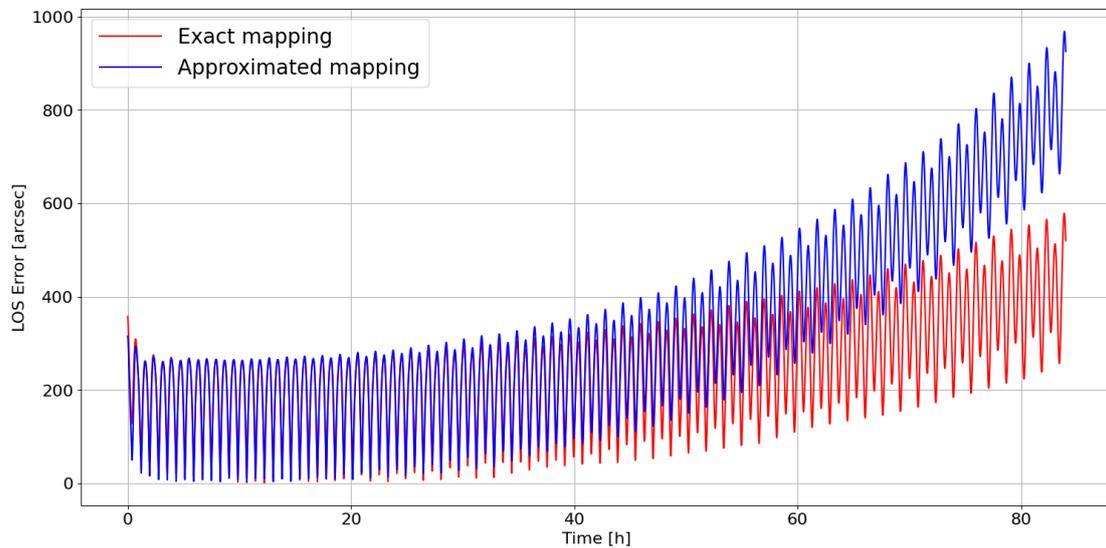
While still not as precise as the numerical propagation, the accuracy is improved by approximately a factor 6 with respect to the rectilinear model. Keeping in mind the unbridgeable gap between the analytical models and the numerical propagation, it is possible to state that, even if not extremely accurate, the  $J_2$  and Drag relative motion model in curvilinear coordinates is suited for onboard angles-only navigation. As said, the exact mapping from CCF to RTN has been here considered in order to evaluate the performance of the transformation. It would be nice however to see how precise is the approximated transformation with respect to its exact counterpart, in order to understand if it would be actually possible to use this simpler, more straightforward conversion for the IROD problem.

Recalling the differences between the approximated and exact transformations, it should be noticed how the most important assumptions that have been made are the two angles  $\hat{\phi}$  and  $\hat{\theta}$  being small, with the consequent trigonometric functions approximations. While this is in general not the case, it can be assumed as true when dealing with a space rendezvous problem: indeed, it would not make sense to consider the relative dynamics of two spacecraft, if they were not close enough to at least hypothesize the possibility of performing a rendezvous manoeuvre. To give an idea, for the use case that has been considered so far, the  $a\delta\lambda$  value of 10 km implies an angular separation in the orbital plane of approximately  $0.08^\circ$ , at 500 km of altitude. The same holds true for the out of plane separation: orbits indeed shall be at least quasi-coplanar, since the change of inclination in LEO is known to be one of the most fuel-intensive orbital manoeuvres. In the use case, the value of  $\hat{\phi}$  in degrees is approximately  $2 \cdot 10^{-5}$ . This also justifies the second assumption that has been made, that is the higher order of magnitude of  $\theta$  with respect to  $\phi$ . In Figure 4.19, the comparison between the two transformations is depicted, for the use case analyzed. Since the two plots follow almost the same trend, it has been decided to keep them in two separate windows, because doing otherwise would have resulted in a superimposition of one graph above the other.



**Figure 4.19:** Comparison between exact and approximate curvilinear transformation

This, however, may not be always the case: as a matter of fact, the two spacecraft may be separated by a much smaller distance in the along-track direction, or may be located in two orbital planes with a slightly higher relative tilt compared to the ones considered so far. Considering for instance a value for  $a\delta i_x$  of  $-1 \text{ km}$  would translate into an inclination difference between Target and Chaser of approximately  $0.01^\circ$ , at  $500 \text{ km}$  of altitude. The  $\hat{\phi}$  angle can be approximated for simplicity with the inclination difference between the two orbits. Having now comparable values for  $\phi$  and  $\theta$ , it is intended to investigate what are the differences, if any, between the exact and the approximated transformations. The Figure 4.20 shows how actually there exists a quite remarkable difference between the two transformations, with the approximated one being less precise and accurate than the exact one. Nevertheless, for a rendezvous mission the Chaser is usually required to be in the orbital plane of the Target, in order to initiate the approach: this implies that, for the scope of this Thesis, the two transformations can be used interchangeably, without loss of generality and accuracy.



**Figure 4.20:** Accuracy difference between the two transformations



# 5

## Initial Relative Orbit Determination

This Chapter intends to provide an extensive and exhaustive description of the IROD problem and of the IROD algorithm that has been implemented as part of the Thesis work. Given the requirements that the algorithm shall satisfy, several of the options mentioned in Section 1.5 had to be discarded. Therefore, in the context of the Thesis work, it has been decided to implement and evaluate the IROD algorithm proposed in [16].

### 5.1 Observability enhancement through CCF

It has been previously demonstrated how employing a curvilinear set of coordinates drastically improves the LOS accuracy of the analytical relative motion models, making them more suited for onboard relative navigation purposes. However, how these coordinates affect the observability of the angles-only navigation is yet to be demonstrated. It has described before how making the problem of angles-only navigation observable implies the possibility to find a unique solution that matches a certain measurement profile or, in short, a  $\mathbf{x}_0$  that uniquely satisfies:

$$\mathbf{u}_i \times \mathbf{r}_i(t_i, \mathbf{x}_0) = 0, \quad i = 1, \dots, n. \quad (5.1)$$

So far,  $\mathbf{r}_i$  has been retrieved through a linear formulation, specifically

$$\mathbf{r}_i = \mathbf{C}(t_i)\Phi(t_i, t_0)\mathbf{x}_0, \quad (5.2)$$

which as previously results in an infinite number of possible solutions to Equation 5.1. Therefore, it would be beneficial to retrieve  $\mathbf{r}_i$  through a non-linear function  $g$  which determines a unique value for  $\mathbf{r}_i$  given an initial state vector  $\mathbf{x}_0$ . The mapping function from the curvilinear to the rectilinear system can indeed fulfil the requirements. Both in the exact and approximated forms, it introduces some non-linear terms that determine the unicity of the solution to Equation 5.1. Writing:

$$\mathbf{u}_i \times g(\mathbf{C}(t_i)\Phi(t_i, t_0)\mathbf{x}_0) = 0 \quad (5.3)$$

with  $g$  representing the aforementioned mapping (Equation 4.8) highlights how, given an arbitrary positive scaling factor  $\mu$  such that the initial condition is  $\mu\mathbf{x}_0$ , the equation is solved by two distinct and independent initial state vectors:

$$g(\mathbf{C}(t_i)\Phi(t_i, t_0)\mathbf{x}_0) \neq g(\mathbf{C}(t_i)\Phi(t_i, t_0)\mu\mathbf{x}_0). \quad (5.4)$$

It shall be recalled now that the most suited analytical model for angles-only navigation has been shown to be the  $J_2$  and Drag model (Section 2.2.6), with the correction introduced by the curvilinear coordinates transformation. This model propagates a 7-dimensional state vector representing the so-called *augmented* ROEs state, which includes the 6 base ROEs plus  $a\delta\dot{a}$ . This means that the matrix  $\mathbf{C}(t)$  defined in Equation 3.11 is not applicable, being it a 6-columns matrix. Therefore from now on, when mentioning  $\mathbf{C}(t)$ , the following formulation will be considered:

$$\mathbf{C}(t) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & -\cos u & -\sin u \\ 0 & 0 & 1 & 0 & 0 & 2\sin u & -2\cos u \\ 0 & 0 & 0 & \sin u & -\cos u & 0 & 0 \end{pmatrix}, \quad (5.5)$$

which highlights how  $a\delta\dot{a}$  does not contribute to the computation of the relative position vector  $\mathbf{r}$  from the augmented ROEs state.

Overall, while the improvements on the observability may not be as drastic as they would be with other techniques like the execution of manoeuvres, they are still decent enough to allow performing the Initial Relative Orbit Determination, while at the same time avoiding the waste of precious resources like fuel.

## 5.2 Algorithm for IROD

The core underlying principle around which the IROD algorithm is built is that the solution to the IROD problem  $\mathbf{x}_0$  is the one that minimizes the already mentioned cost function  $J$ , here recalled for completeness:

$$J = \sum_{i=1}^n \|\mathbf{u}_i \times \mathbf{h}(t_i, \mathbf{x}_0)\|^2. \quad (5.6)$$

While several techniques for the minimum search may be exploited, the main difficulty is the computation of a solution in a reasonable amount of time, given the limited onboard processing power.

It has already been discussed extensively how a linear motion model cannot be used to perform the IROD, given the intrinsic non observability of the linear formulation of the dynamics and of the measurement model; however the solution coming from the linear theory, which will be covered extensively in the next Section, can be used to initialize a batch least-squares adjustment in the vicinity of the true solution. To do so, it is necessary to scale the solution coming from the linear theory, called  $\hat{\mathbf{x}}_0$ , by a set of scaling factors  $\mu$ , that can be chosen arbitrarily or via iterative refinements. Naming  $\hat{\mathbf{x}}_0^\mu$  the scaled initial state vector coming from the linear model, and  $\sigma(\mu)$  the root-mean-square value of the measurements fitting residuals associated to  $\hat{\mathbf{x}}_0^\mu$ , so that:

$$\sigma(\mu) = \sqrt{\frac{1}{n}J(\hat{\mathbf{x}}_0^\mu)}, \quad (5.7)$$

it is possible to extract a curve which is function of  $\mu$ , whose minimum is given by the value  $\hat{\mu}$  which represents the best estimate of the scaling factor of the solution.

### 5.2.1 Linear motion model solution

As said, the first step to be performed is the computation of the solution  $\hat{\mathbf{x}}_0$  associated to the linear motion model. To do so, it is desirable to first write Equation 5.1 in a more compact form:

$$\mathbf{U}_i \mathbf{C}(t_i) \Phi(t_i, t_0) \hat{\mathbf{x}}_0 = \mathbf{A}_i \hat{\mathbf{x}}_0 = 0, \quad (5.8)$$

where  $\mathbf{U}_i$  is the cross-product matrix associated with the  $i$ -th LOS measurement vector, expressed in the RTN frame. Following the acquisition of  $n$  measurements, the matrix  $\mathbf{A}$  is given by:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_n \end{pmatrix}, \quad \text{for } i = 1, \dots, n. \quad (5.9)$$

Recalling the definitions of the matrices  $\mathbf{C}(t_i)$  and  $\Phi(t_i, t_0)$ , it is evident that the algorithm requires data from the Chaser's GPS sensors to function as intended. Specifically, the Chaser's state at the initial time  $t_0$  is required to compute the State-Transition Matrix, while its state at each  $i$ -th measurement is necessary to compute both the STM and the conversion matrix  $\mathbf{C}$  which transforms the ROE state into the RTN position. Considering the 7-dimensional augmented ROEs state, given  $n$  measurements  $\mathbf{A}$  is a matrix of  $3n \times 7$  dimension. Now, considering a columns-wise partition of  $\mathbf{A}$  such that:

$$\mathbf{A} = (\mathbf{A}_1 \ \mathbf{A}_2 \ \mathbf{A}_3 \ \mathbf{A}_4 \ \mathbf{A}_5 \ \mathbf{A}_6 \ \mathbf{A}_7), \quad (5.10)$$

the linear system can be equivalently written as:

$$\tilde{\mathbf{A}} \hat{\mathbf{x}}_0 = -\mu \mathbf{A}_3 \quad (5.11)$$

with

$$\begin{aligned} \tilde{\mathbf{A}} &= (\mathbf{A}_1 \ \mathbf{A}_2 \ \mathbf{A}_4 \ \mathbf{A}_5 \ \mathbf{A}_6 \ \mathbf{A}_7) \\ \hat{\mathbf{x}}_0 &= (a\delta\dot{a} \ a\delta a \ a\delta i_x \ a\delta i_y \ a\delta e_x \ a\delta e_y)^T \end{aligned} \quad (5.12)$$

and  $\mu = a\delta\lambda$ . If  $\mu$  is set equal to 1, then the state vector  $\hat{\mathbf{x}}_0$  is in its unscaled form. The scaled version for a generic value of  $\mu$  is obtained as:

$$\hat{\mathbf{x}}_0^\mu = \mu \hat{\mathbf{x}}_0. \quad (5.13)$$

The choice of  $a\delta\lambda = \mu$  as scaling factor is determined by the fact that a physical range of possible values for this variable can be more intuitively derived, with respect

to the other ROEs. Indeed, it is necessary to have a pre-defined search space to find actual  $\mu$  value, as it will be highlighted in the next Sections. Two things are worth mentioning: first, the scaling factor  $\mu$  represents a distance in meters. Therefore, setting its value to 1 to retrieve the unscaled initial ROEs vector corresponds to assigning a value of 1 meter. Second, the full initial vector  $\hat{\mathbf{x}}_0^\mu$ , which includes all the 7 initial ROEs estimated from the linear model, contains  $\mu$  in the position of  $a\delta\lambda$ . In other words,  $a\delta\lambda$  is set equal to  $\mu$ .

The solution can be computed following different approaches. Specifically, two possible methods have been exploited, and they are briefly presented here.

The first is a Singular Value Decomposition (SVD), which breaks down the matrix  $\tilde{\mathbf{A}}$  is the form:

$$\tilde{\mathbf{A}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (5.14)$$

where  $\mathbf{U}$  is a matrix containing the left-singular vectors of  $\tilde{\mathbf{A}}$ ,  $\mathbf{\Sigma}$  is a diagonal matrix containing the singular values of  $\tilde{\mathbf{A}}$  and  $\mathbf{V}$  contains the right-singular vectors. Calling  $\mathbf{b} = -\mu\mathbf{A}_3$ , the solution to the linear system is given by (intermediate steps omitted):

$$\hat{\mathbf{x}}_0 = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{b}. \quad (5.15)$$

The second approach instead retrieves the solution in a least-squares sense, so that:

$$\hat{\mathbf{x}}_0 = (\tilde{\mathbf{A}}^T\tilde{\mathbf{A}})^{-1}\tilde{\mathbf{A}}^T\mathbf{b}. \quad (5.16)$$

In applying this method, the great advantage comes from the block structure of  $\tilde{\mathbf{A}}$  and  $\mathbf{b}$ : those are in fact composed by blocks of size  $3 \times 6$  and  $3 \times 1$  called  $\mathbf{A}_i$  and  $\mathbf{b}_i$  respectively, each associated to a specific measurement and a to specific epoch  $t_i$ . This allows to compute the terms  $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$  and  $\tilde{\mathbf{A}}^T\mathbf{b}$ , very efficiently, in facts:

$$\begin{aligned} \tilde{\mathbf{A}}^T\tilde{\mathbf{A}} &= \sum_{i=1}^n \tilde{\mathbf{A}}_i^T\tilde{\mathbf{A}}_i, \\ \tilde{\mathbf{A}}^T\mathbf{b} &= \sum_{i=1}^n \tilde{\mathbf{A}}_i^T\mathbf{b}_i. \end{aligned} \quad (5.17)$$

This formulation turns out to be extremely convenient when dealing with large systems, whose dimension is the consequence of the acquisition of a massive number of measurements: while the computational cost of SVD increases significantly as the system size grows, the cost of the least-squares-based approach remains nearly constant and it is primarily determined by the multiplication of very small matrices, their inversion, and their summation. Lowering the computational cost as much as possible is a critical aspect that from now on will be constantly addressed in the Thesis: it shall not be forgotten that one of the goals of the development of the IROD method is the realization of an algorithm which satisfies the requirements for onboard implementation. This means that the runtime must be low enough that, when deployed on a real satellite processor, the algorithm can still find a solution within a reasonable amount of time. A quantitative analysis of the runtime for the

whole algorithm will be provided later on in this Thesis, while for now a comparison between the two methods for the sole linear model solution will be presented.

As mentioned, the parameter that determines the system size is the number of acquired measurements. This parameter, which will be quantified later, must not be too low or too high: too few measurements may prevent computing a correct solution, while too many could lead to an excessively heavy computational process. For the  $\tilde{\mathbf{A}}$  matrix, the blocks associated to a single measurement have size  $3 \times 6$ : this means, that, being  $n$  the number of measurements acquired, the size of the system will be  $3n \times 6$ . In the SVD, for a system with  $n \gg p$ , the computational cost grows linearly with the number of rows, and it tends to  $O(np^2)$ . The least-squares based method instead relies on the fact that, regardless of the number of measurements acquired, the system to be solved will always be  $6 \times 6$ , with an extra small cost related to the inversion of  $\tilde{\mathbf{A}}^T \tilde{\mathbf{A}}$ . One may argue that the costs associated to the transposition of  $\tilde{\mathbf{A}}$  and, most important, with the addition of the  $\tilde{\mathbf{A}}_i^T \tilde{\mathbf{A}}_i$  terms has not been considered: nevertheless, it should be remembered that each matrix corresponds to single LOS measurement, acquired with a certain frequency that leaves time to the algorithm to perform the aforementioned operations during the acquisition, rather than afterwards. Once all measurements are collected, the only thing left to do is indeed the computation of the system solution. The table below shows the differences between the two methods, for several number of acquired LOS measurements.

	<b>Computation Time [<math>\mu</math>s]</b>	
<b>Measurements</b>	<b>SVD</b>	<b>Least-Squares</b>
500	98	12
1000	236	12
2000	551	12
4000	1054	13

**Table 5.1:** Computation time of linear model solution with SVD and least-squares on Intel Core i9-13950HX 2.20 GHz

Obviously, the runtime varies consistently depending on the processor on which the C++ code is executed. In this case, the processor is an Intel Core i9-13950HX 2.20 GHz. The table shows how the computation time for the SVD grows approximately linearly with the number of measurements, while the least-squares approach runs always in the same time.

As previously discussed, LOS measurements consist of a set of two angles, azimuth and elevation, acquired through an optical camera. In this phase, however, the algorithm is fed with LOS measurements that are based on the actual LOS angles, computed by considering only the true trajectories of the Chaser and Target, which are propagated using the high fidelity simulator. Therefore, errors and uncertainties related to image processing, centroid filtering, or other types of sensor noises are not modeled or accounted for in this phase. As described in Chapter 3, from the LOS angles it is possible to extract the normalized LOS vector in the RTN frame, whose components are the entries of the matrix  $\mathbf{U}_i$ .

### 5.2.2 Non-linear model solution

At this point, it is possible to start the search of the non-linear motion model solution, which ideally represents the true, actual relative state in terms of ROEs between the two spacecraft. The non-linear model employed will be the  $J_2$  and Drag analytical model (Section 2.2.6) with the addition of the non-linear mapping from curvilinear to RTN coordinates. In particular, given the hypotheses made in Chapter 4, it is safe to affirm that the approximate transformation is suited for the IROD method. Recalling that:

$$\mathbf{r} \approx \check{\mathbf{r}} + \begin{pmatrix} -\frac{\theta^2}{2R} \\ 0 \\ 0 \end{pmatrix}, \quad (5.18)$$

where for clarity the new notation  $R$  is being used to denote the Chaser's orbit radius, it is possible to write Equation 5.1 as:

$$\mathbf{u}_i \times \left( \check{\mathbf{r}}_i - \frac{\theta_i^2}{2R} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right) = 0, \quad i = 1, \dots, n. \quad (5.19)$$

It is now convenient, for clarity and compactness, to introduce the matrix  $\mathbf{D}_i$ , defined as:

$$\mathbf{D}_i = \mathbf{C}_i \Phi(t_i, t_0), \quad (5.20)$$

and recalling that  $\theta$  represents the second component of the relative position vector in CCF, Equation 5.19 can be rewritten as:

$$\mathbf{U}_i \left( \mathbf{D}_i \mathbf{x}_0 - \frac{1}{2R} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \mathbf{x}_0^T \mathbf{D}_{i2,1-7}^T \mathbf{D}_{i2,1-7} \mathbf{x}_0 \right) = 0, \quad (5.21)$$

where  $\mathbf{D}_{i2,1-7}$  stands for the second row of  $\mathbf{D}_i$ .

The idea now is to perform a series of local minimizations around the linear model solution  $\hat{\mathbf{x}}_0^\mu$ , each of them with a different scaling factor  $\mu$ . It is legitimate to assume that the solution of each minimization will be close to  $\hat{\mathbf{x}}_0^\mu$ . Naming

$$\mathbf{q}_i(\mathbf{x}) = -\frac{1}{2R} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \mathbf{x}^T \mathbf{D}_{i2,1-7}^T \mathbf{D}_{i2,1-7} \mathbf{x}, \quad (5.22)$$

linearizing Equation 5.21 around  $\hat{\mathbf{x}}_0^\mu$  yields:

$$\mathbf{U}_i \left( \mathbf{D}_i \mathbf{x}_0 + \mathbf{q}_i(\hat{\mathbf{x}}_0^\mu) + \left. \frac{d\mathbf{q}_i}{d\mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_0^\mu} (\mathbf{x}_0 - \hat{\mathbf{x}}_0^\mu) \right) = 0. \quad (5.23)$$

Considering now the linear model solution decomposition in  $(\hat{\mathbf{x}}_0, \mu)$ , reminding that  $\mu$  replaces  $a\delta\lambda$  at the third position in the state vector, it is possible to obtain the following relation:

$$\mathbf{U}_i \left( \tilde{\mathbf{D}}_i - \frac{1}{R} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \hat{\mathbf{x}}_0^{\mu T} \mathbf{D}_{i2,1-7}^T \tilde{\mathbf{D}}_{i2,1-6} \right) \cdot \tilde{\mathbf{x}}_0 = \mathbf{U}_i \left( -\mathbf{D}_{i,3} \mu + \frac{1}{R} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \hat{\mathbf{x}}_0^{\mu T} \mathbf{D}_{i2,1-7}^T \mathbf{D}_{i23} \mu + \mathbf{q}_i(\hat{\mathbf{x}}_0^{\mu T}) \right), \quad (5.24)$$

which presents the structure of a linear system in the form  $\mathbf{A}_i \tilde{\mathbf{x}}_0 = \mathbf{b}_i$ . For clarity,  $\mathbf{D}_{i,3}$  refers to the third column of  $\mathbf{D}_i$  or, in other words, the column that subtracted from  $\mathbf{D}_i$  turns it into  $\tilde{\mathbf{D}}_i$ . Furthermore,  $\mathbf{D}_{i23}$  is the element of  $\mathbf{D}_i$  in position (2, 3). After acquiring  $n$  measurements, the system takes the form  $\mathbf{A}(\mu) \tilde{\mathbf{x}}_0 = \mathbf{b}(\mu)$ ,  $\mathbf{A}$  being a  $3n \times 6$  matrix. At this point it should be noted that this strategy would lead to the calculation of the minimum of the function  $J = \sum_{i=1}^n \|\mathbf{u}_i \times \mathbf{r}(t_i, \mathbf{x}_0)\|^2$ , while the goal is to find the minimum of  $J = \sum_{i=1}^n \|\mathbf{u}_i \times \mathbf{h}(t_i, \mathbf{x}_0)\|^2$ . Therefore, for the sake of consistency, it is necessary to divide both sides of Equation 5.24 by the norm of the relative position vector associated with the scaled linear solution:

$$\|\mathbf{r}^\mu\| = \|\mu \mathbf{C}(t_0) \hat{\mathbf{x}}_0\| \quad (5.25)$$

The minimum of the system, introducing  $\hat{\mathbf{x}}_0^\mu$ , can be now derived as:

$$m(\mu) = \|\mathbf{A}(\mu) \hat{\mathbf{x}}_0^\mu - \mathbf{b}(\mu)\|^2, \quad (5.26)$$

which can be rewritten as:

$$m(\mu) = \mathbf{b}(\mu)^T \mathbf{M}(\mu) \mathbf{b}(\mu), \quad (5.27)$$

where  $\mathbf{M}(\mu) = \mathbf{I} - \mathbf{P}(\mu)$ , with  $\mathbf{P} = \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ , projection matrix of  $\mathbf{A}$ .

Given the structure of  $\mathbf{b}$ , highlighted in Equation 5.24, it is possible to decompose it in two terms, such that:

$$\begin{aligned} \mathbf{b}_1 &= \frac{1}{\|\mathbf{r}^\mu\|} \mathbf{U} \left( -\mu \mathbf{D}_{:,3} + \frac{1}{R} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \mu^2 \hat{\mathbf{x}}_0^T \mathbf{D}_{2,1-7}^T \mathbf{D}_{23} \right) \\ \mathbf{b}_2 &= \frac{1}{\mu \|\mathbf{r}^\mu\|} \mathbf{U} \left( -\frac{1}{2R} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \mu^2 \hat{\mathbf{x}}_0^T \mathbf{D}_{2,1-7}^T \mathbf{D}_{2,1-7} \hat{\mathbf{x}}_0 \right), \end{aligned} \quad (5.28)$$

so that  $\mathbf{b} = (\mathbf{b}_1 + \mu \mathbf{b}_2)$ . This way,  $m(\mu)$  can be rewritten as:

$$m(\mu) = \mathbf{b}_1^T \mathbf{M} \mathbf{b}_1 + \mu^2 \mathbf{b}_2^T \mathbf{M} \mathbf{b}_2 + 2\mu \mathbf{b}_1^T \mathbf{M} \mathbf{b}_2. \quad (5.29)$$

Given the properties of the projection matrix,  $\mathbf{M}$  is semi-definite positive. This implies that the function is convex and a minimum can be found at:

$$\hat{\mu} = -\frac{\mathbf{b}_1^T \mathbf{M} \mathbf{b}_2}{\mathbf{b}_2^T \mathbf{M} \mathbf{b}_2}. \quad (5.30)$$

After the true scaling factor  $\hat{\mu}$  is found, it is possible to compute the remaining elements of the initial state vector,  $\tilde{\mathbf{x}}_0$ , by solving the system  $\mathbf{A}(\hat{\mu}) \tilde{\mathbf{x}}_0 = \mathbf{b}(\hat{\mu})$ . Denying this last step would mean not accounting for the effect of the curvature on the other ROEs. Scaling the linear motion model solution by  $\hat{\mu}$ , although viable given the small discrepancies that would result, is not consistent with the approach and the methodology considered so far.

### 5.2.3 Structure of the algorithm

While for the linear motion model solution the implementation in the form of C++ code is pretty straightforward and totally resembles what has been described in Section 5.2.1, care has to be taken in highlighting the differences and optimizations made in the implementation of the non-linear model solution. As a matter of fact, it should always be kept in mind that the requirement for low computational intensity is a constraint that must be addressed carefully.

The matrices  $\mathbf{U}_i$  and  $\mathbf{D}_i$  are computed during the measurements acquisition and then stored for the further use during the non-linear model solution computation. From those, and from the linear model solution, it is possible to compute the terms that compose Equation 5.24. For clarity and conciseness, from now on they will be referred as follows:

$$\begin{aligned}
 \mathbf{L}_{1_i} &= \mathbf{U}_i \tilde{\mathbf{D}}_i \\
 \mathbf{L}_{2_i} &= -\mathbf{U}_i \frac{1}{R} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \hat{\mathbf{x}}_0^T \mathbf{D}_{i2,1-7}^T \tilde{\mathbf{D}}_{i2,1-6} \\
 \mathbf{R}_{1_i} &= -\mathbf{U}_i \mathbf{D}_{i,3} \\
 \mathbf{R}_{2_i} &= \mathbf{U}_i \frac{1}{R} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \hat{\mathbf{x}}_0^T \mathbf{D}_{i2,1-7}^T \mathbf{D}_{i23} \\
 \mathbf{R}_{3_i} &= -\mathbf{U}_i \frac{1}{2R} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \hat{\mathbf{x}}_0^T \mathbf{D}_{i2,1-7}^T \mathbf{D}_{i2,1-7} \hat{\mathbf{x}}_0^T.
 \end{aligned} \tag{5.31}$$

For  $n$  measurements, this results in:

$$\mathbf{L}_1 = \begin{pmatrix} \mathbf{L}_{1_1} \\ \mathbf{L}_{1_2} \\ \vdots \\ \mathbf{L}_{1_n} \end{pmatrix}, \quad \mathbf{R}_1 = \begin{pmatrix} \mathbf{R}_{1_1} \\ \mathbf{R}_{1_2} \\ \vdots \\ \mathbf{R}_{1_n} \end{pmatrix}.$$

For brevity,  $\mathbf{L}_2$ ,  $\mathbf{R}_2$  and  $\mathbf{R}_3$  have been omitted.

For a certain scaling factor  $\mu$ , the following relations ultimately are obtained and hold true:

$$\begin{aligned}
 \mathbf{A}(\mu) &= \frac{1}{\|\mathbf{r}^\mu\|} (\mathbf{L}_1 + \mu \mathbf{L}_2) \\
 \mathbf{b}_1(\mu) &= \frac{1}{\|\mathbf{r}^\mu\|} (\mu \mathbf{R}_1 + \mu^2 \mathbf{R}_2) \\
 \mathbf{b}_2(\mu) &= \frac{1}{\mu \|\mathbf{r}^\mu\|} (\mu^2 \mathbf{R}_3).
 \end{aligned} \tag{5.32}$$

On a computational effort perspective, the real bottleneck of the algorithm is represented by the computation of the matrix  $\mathbf{M}$ . This is indeed a  $3n \times 3n$  matrix obtained as:

$$\mathbf{M} = \mathbf{I} - \mathbf{P} = \mathbf{I} - \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T, \tag{5.33}$$

which is a fairly complex operation in terms of computational cost. In order to reduce the effort, it is possible to leverage once again the block structure of  $\mathbf{A}$ . In particular, it is possible to alleviate the cost associated to the computation of  $(\mathbf{A}^T \mathbf{A})^{-1}$ , by pre-computing some terms that are not dependent on  $\mu$ , allowing their reusability. Specifically:

$$\begin{aligned}\mathbf{T}_1 &= \sum_{i=1}^n \mathbf{L}_{1_i}^T \mathbf{L}_{1_i} \\ \mathbf{T}_2 &= \sum_{i=1}^n (\mathbf{L}_{1_i}^T \mathbf{L}_{2_i} + \mathbf{L}_{2_i}^T \mathbf{L}_{1_i}) \\ \mathbf{T}_3 &= \sum_{i=1}^n \mathbf{L}_{2_i}^T \mathbf{L}_{2_i}.\end{aligned}\tag{5.34}$$

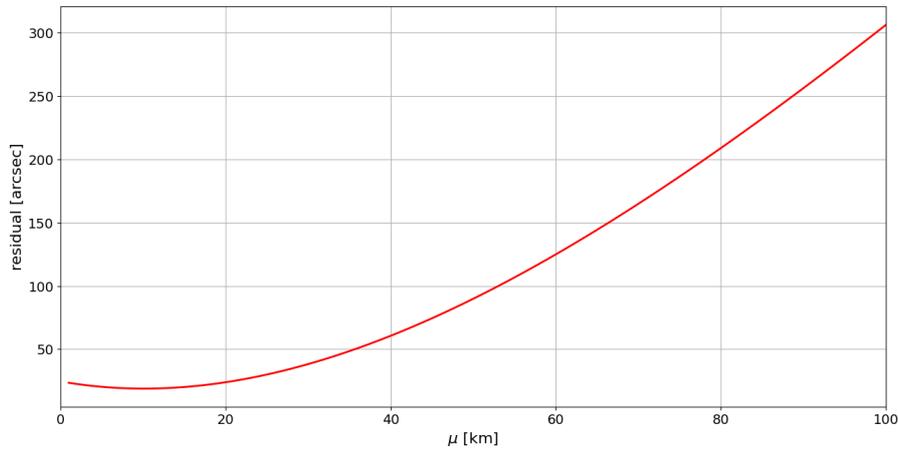
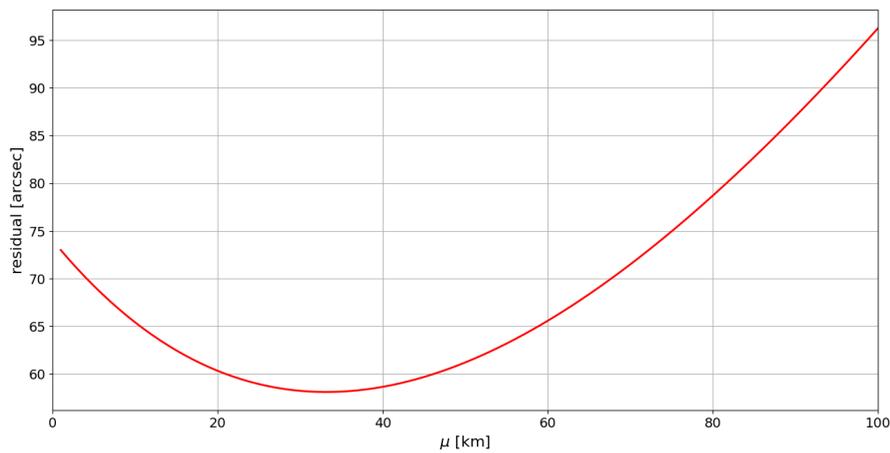
This allows to compute  $\mathbf{A}^T \mathbf{A}$  for each  $\mu$  as:

$$\mathbf{A}^T \mathbf{A} = \frac{1}{\|\mathbf{r}^\mu\|^2} (\mathbf{T}_1 + \mu \mathbf{T}_2 + \mu^2 \mathbf{T}_3).\tag{5.35}$$

While this light optimization certainly gives its contribution, the computational cost associated with  $\mathbf{M}$  is still considerably high. To give an idea, even with the pre-computation of the terms associated with  $(\mathbf{A}^T \mathbf{A})^{-1}$ , the total cost of the operation is  $O(n^2)$ , which becomes critical when the number of measurements starts exceeding the thousand. The main problem here is not just the cost of the single operation, but rather the number of times this operation has to be executed. As a matter of fact, if one wants to build the curve  $m(\mu)$  appropriately, several iterations for different values of  $\mu$  are needed, each of them requiring a new computation of  $\mathbf{M}$ . Specifically, the inter-satellite separation can vary significantly depending on the type of mission and the phase of the rendezvous manoeuvre in which the two spacecraft are engaged. Therefore, the pool of values from which to choose  $\mu$  from cannot be too small, as this would significantly increase the risk of missing the true scaling factor. At the same time, to properly evaluate the function  $m(\mu)$ , its shape, and the location of its minimum, the interval from which the scaling factors are selected must be densely subdivided. In order to take all these factors into account, a reasonable choice could be considering an interval such that  $[\mu_A; \mu_B] = [1; 100] km$ , with a granularity of  $1 km$ . This results in a total of 100 iterations during which  $\mathbf{M}$  is computed, leading to the previously mentioned computational cost issues.

To overcome this, it is necessary to find a method capable of finding the location of the minimum of  $m(\mu)$  with a low amount of iterations. The shape of the function, which is convex and has one single global minimum (Figure 5.1), makes the application of a simple method like the binary search ideal for the scope. As a matter of fact, if the initial size of the interval is  $100 km$ , 10 iterations are sufficient to scale it of a factor  $2^{10}$ , bringing its size to less than  $100 m$ . The binary search algorithm is executed as follows: first, an initial search interval is defined, for instance  $[\mu_A; \mu_B] = [1; 100] km$ ; after that, the derivative of  $m(\mu)$  is evaluated at  $\mu_A$  and  $\mu_K = \frac{1}{2}(\mu_A + \mu_B)$ :

$$m'(\mu) = 2\mathbf{b}_1^T \mathbf{M} \mathbf{b}_2 + 2\mu \mathbf{b}_2^T \mathbf{M} \mathbf{b}_2.\tag{5.36}$$

(a) Residual curve for an initial along-track distance of 10 *km*(b) Residual curve for an initial along-track distance of 30 *km***Figure 5.1:** Residual curves for initial along-track distances of (a) 10 *km* and (b) 30 *km*

If  $m'(\mu_A) \cdot m'(\mu_K) \leq 0$ , the slope of the function changes sign within  $[\mu_A; \mu_K]$ , therefore the minimum must be located within this interval. On the contrary, if  $m'(\mu_A) \cdot m'(\mu_K) > 0$ , the minimum must be located in  $[\mu_K; \mu_B]$ . The process is repeated starting from the newly obtained interval, until the size of the interval falls below a used-defined threshold. As said, with the initial interval  $[1; 100]$  *km* and a threshold of 100 *m*, 10 iterations are executed before the algorithm stops. For the extreme point of the interval retained for the next iteration, the value of  $m'(\mu)$  is cached to save computation time by avoiding its recomputation. Once the size of the interval drops under the threshold, its middle point is assumed to be the value of  $\mu$  for which  $m(\mu)$  reaches its minimum. Without further refinements, this means that the discrepancy between the assumed minimum and the actual minimum will never exceed 50 *m*. Overall, this approach contributes in reducing the time of execution of the IROD algorithm by approximately a factor 10, with respect to the case in which  $m(\mu)$  is evaluated in all the points of the interval with 1 *km* granularity. The

last step that must be performed is the computation of the true scaling factor  $\hat{\mu}$ , which is function of the  $\mu$  value which minimizes the curve. This step is done more for preciseness and rigour rather than for necessity, since  $\mu$  is expected to be already quite close to the true solution. However, as Equation 5.30 states,  $\mu$  needs to be used to compute the correspondent  $\mathbf{M}$ ,  $\mathbf{b}_1$  and  $\mathbf{b}_2$  values that return  $\hat{\mu}$ .

The value of  $\hat{\mu}$  represents the "real" value of  $a\delta\lambda$ , computed accounting for the curvature of the orbital path. As mentioned before, the full initial ROEs can be found solving the system  $\mathbf{A}(\hat{\mu})\tilde{\mathbf{x}}_0 = \mathbf{b}(\hat{\mu})$ . Just like how it has been done for the linear model solution, it is possible to solve the system in the least-squares sense as:

$$\tilde{\mathbf{x}}_0 = (\mathbf{A}(\hat{\mu})^T \mathbf{A}(\hat{\mu}))^{-1} \mathbf{A}(\hat{\mu})^T \mathbf{b}(\hat{\mu}). \quad (5.37)$$

For  $\mathbf{A}(\hat{\mu})^T \mathbf{A}(\hat{\mu})$ , one can use once again Equation 5.38. Analogously, for  $\mathbf{A}(\hat{\mu})^T \mathbf{b}(\hat{\mu})$  it is possible to precompute some terms in order to avoid the transposition of  $\mathbf{A}(\hat{\mu})$ . Hence, simultaneously at the computation of  $\mathbf{T}_1$ ,  $\mathbf{T}_2$ ,  $\mathbf{T}_3$ , the following operations are performed:

$$\begin{aligned} \mathbf{B}_1 &= \sum_{i=1}^n \mathbf{L}_{1_i}^T \mathbf{R}_{1_i} \\ \mathbf{B}_2 &= \sum_{i=1}^n \mathbf{L}_{1_i}^T \cdot (\mathbf{R}_{2_i} + \mathbf{R}_{3_i}) \\ \mathbf{B}_3 &= \sum_{i=1}^n \mathbf{L}_{2_i}^T \mathbf{R}_{1_i} \\ \mathbf{B}_4 &= \sum_{i=1}^n \mathbf{L}_{2_i}^T \cdot (\mathbf{R}_{2_i} + \mathbf{R}_{3_i}). \end{aligned} \quad (5.38)$$

This allows to express  $\mathbf{A}(\hat{\mu})^T \mathbf{b}(\hat{\mu})$  as:

$$\mathbf{A}(\hat{\mu})^T \mathbf{b}(\hat{\mu}) = \hat{\mu} \mathbf{B}_1 + \hat{\mu}^2 (\mathbf{B}_2 + \mathbf{B}_3) + \hat{\mu}^3 \mathbf{B}_4. \quad (5.39)$$

The total runtime of the entire algorithm is obviously influenced remarkably by the number of measurements that are acquired, just like the computation of the linear model solution. Interestingly, it is possible to observe a increasing computation time trend which is roughly proportional to  $n^2$ . This clearly demonstrates how the runtime is primarily influenced by the computational effort required to calculate  $\mathbf{M}$ , with all other operations having a negligible impact on the overall duration.

Measurements	Computation time [s]
500	0.2
1000	0.9
2000	3.2
4000	12.4

**Table 5.2:** Computation time for IROD on Intel Core i9-13950HX 2.20 GHz

### 5.3 Preliminary performance assessment

This section intends to provide an insight on the performance of the IROD algorithm, under a handful of initial conditions and employed satellites. All these simulations are conducted by taking into account the visibility constraint; furthermore, they consider always the same camera parameters, specifically a Sun Exclusion Angle of  $30^\circ$  and a FOV of  $20^\circ$  both for the horizontal and vertical one.

The first example showcased here refers to the residual curve on Figure 5.1a. The satellites involved are BIROS and BEESAT-4, and they are placed on a generic LEO orbit at an altitude of  $500 \text{ km}$ . This particular scenario is somewhat challenging, given that the visibility is such that the Target is visible only for approximately 10% of the time. The IROD is performed by acquiring a total of  $n = 1500$  LOS measurements spaced from one another by  $\Delta t = 20 \text{ s}$ . The table below provides the results of the orbit determination process, as well as the true initial relative state between the two spacecraft.

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	-3.27e-4	-38.5	10000	-2.34	240	-13.3	260
<b>IROD</b>	-3.35e-4	-38.1	9712	-1.85	231	-14.5	256
<b>Linear (unscaled)</b>	-4.31e-8	-5.55e-3	1	-2e-4	2.43e-2	-2.28e-3	2.71e-2
<b>Linear (scaled)</b>	-4.31e-4	-55.5	10000	-2.00	243	-22.8	271

**Table 5.3:** Full IROD and linear model solutions compared with the reference,  $10 \text{ km}$  separation

Table 5.3 also reports the so-called linear unscaled and linear scaled solution: As can be observed, the linear solution is not far from the reference when scaled by an appropriate factor, such as the actual  $a\delta\lambda$ . Therefore, by computing the linear unscaled solution (derived from the linear model solution step) and having some knowledge of the approximate relative distance between the objects (*e.g.*, from TLEs), it is possible to multiply this a priori known relative range by the linear unscaled ROEs vector to obtain the linear *scaled* solution. This allows to avoid the execution of the full IROD algorithm, saving time and resources but losing some autonomy in the GNC system. An extensive performance assessment of this shorter approach will be provided in Chapter 6.

At this point, it is worth mentioning an important aspect of what the IROD algorithm does, particularly in relation to its ultimate scope of providing an initial state vector to the navigation filter. The IROD solution that is computed refers to the relative state at the beginning of the observation: however, when the Kalman Filter is initialized, it requires the relative state at the exact moment of initialization or, in other words, the state at the time real-time navigation begins. This means that the IROD solution needs to be propagated from the initial time of observation  $t_0$  to the time  $t_f$  at which the filter is initialized. The propagation can be done once again with an analytical motion model, such that

$$\boldsymbol{\alpha}_f = \boldsymbol{\Phi}(t_f, t_0)\boldsymbol{\alpha}_{IROD}. \quad (5.40)$$

To assess if the propagation still guarantees the accuracy of the IROD solution, it is possible to compare  $\boldsymbol{\alpha}_f$  with the actual relative state at  $t_f$ :

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	-3.27e-4	-51.0	12336	-2.24	241	-10.4	257
<b>IROD</b>	-3.35e-4	-49.2	11993	-1.85	232	-8.73	257

**Table 5.4:** IROD solution propagated to the final observation time, 10 km initial separation

Even though the analytical model introduces small discrepancies due to its inherent limitations in accuracy, the solution remains highly precise compared to the reference. This highlights the viability of this approach for determining the relative state required for the Kalman Filter initialization.

Another example is described hereafter: in this case, the Target is the AISat satellite of DLR [47], while the Chaser is BIROS once again. This time, the two satellite are initially separated by a relative distance of approximately 30 km in the along-track direction. A total of  $n = 2500$  measurements is acquired, with a frequency of 1 measurement every 5 seconds. As done previously, the observation is carried out accounting for the Target visibility constraint. Results are reported in the table below:

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	-4.90e-4	-102	30383	-2.99	246	-2.75	249
<b>IROD</b>	-2.41e-4	-107	33119	-2.77	262	0.40	292

**Table 5.5:** IROD solution compared with the reference, 30 km initial separation

A quick look into these results highlights how, compared to the previous case, the relative range  $a\delta\lambda$  is estimated with less accuracy. However, the relative error between the estimated and the actual value remains under the threshold of 10%, which can be considered a indicative reference benchmark when assessing the performance of the IROD algorithm. Another conclusion that can be preliminary drawn from these results is the somewhat difficult determination of  $a\delta\dot{a}$ : this matter will be thoroughly addressed in the following Chapter, however is it possible to make a few comments over the results obtained here. It is likely that the difficulties in estimating this parameter rise from the different nature of the ROE itself: as a matter of fact,  $a\delta\dot{a}$  is the only ROE which does not express a distance, therefore it is not directly related to the geometry of the formation. Thus, its estimation through LOS measurements, which are intrinsically bounded to the geometry of the problem, is somewhat difficult. In this specific test case, the error between the expected and the estimated differential drag amounts to approximately 0.2 mm/s, which can be considered low but not negligible for the type of problem and the duration of the typical rendezvous missions. For what concerns the propagation up to the current time  $t_f$ , the results are the following:

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	-4.90e-4	-111	33541	-3.33	248	0.92	247
<b>IROD</b>	-2.41e-4	-111	36343	-2.78	263	4.10	292

**Table 5.6:** IROD solution propagated to the final observation time, 30 km initial separation

The last example presented here involves another DLR satellite, Eu:CROPIS [48], as Target, and the usual BIROS satellite as Chaser. In this case, the satellites are placed on a relative orbit which features an along-track distance of approximately 46 km, at an altitude of 510 km from the Earth's surface. A total of  $n = 2000$  LOS measurements are acquired, with an interval of 7 seconds from one another. The output of the IROD algorithm is the following:

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	-3.49e-5	-54.9	46312	135	156	-260	-195
<b>IROD</b>	-4.92e-5	-47.7	50734	148	200	-281	-215

**Table 5.7:** IROD solution compared with the reference, 46 km initial separation

The propagated real and estimated states at the end of the acquisition are instead:

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	-3.49e-5	-54.7	47989	135	155	-257	-198
<b>IROD</b>	-4.92e-5	-48.8	52206	148	200	-278	-219

**Table 5.8:** IROD solution propagated to the final observation time, 46 km initial separation



# 6

## IROD method performance assessment

This chapter intends to provide a comprehensive and exhaustive evaluation of the implemented IROD method. In accordance with the initial requirements, the algorithm performance will be assessed in presence of various sources of noise and errors, analyzing whether and to what extent the noisy solution deviates from the ideal one. Subsequently, a thorough analysis using Monte Carlo simulations will be presented to evaluate performance across various mission scenarios and to determine optimal performance parameters such as the necessary number of measurements and their frequency of acquisition. Following that, the algorithm's performance with real data from a past space mission of DLR will be presented. Finally, the runtime performance will be assessed by deploying and executing the code on a processor which will be used as satellite's onboard computer.

### 6.1 Performance with noise

#### 6.1.1 Sources of noise and errors

In a real mission scenario, the overall performance of the IROD algorithm would be affected by several sources of noise and errors the inherently characterize the hardware and the sensors onboard the satellites.

For the specific task at hand, orbit determination using angles-only measurements, the primary source of error is the camera noise. Specifically, the navigation optical sensor will acquire LOS measurements of the Target which will be affected by errors, that result in discrepancies between the actual and the measured values of the azimuth and elevation angles. These discrepancies depends mostly on the quality of the sensor itself and on the accuracy of the image processing task. The camera used for the far range angles-only navigation in the AVANTI mission of DLR was affected by a maximum LOS error of approximately 40", which translates to an error of 5 meters with 30 *km* of inter-satellite distance [16][49].

As described in Chapter 5, the IROD algorithm requires the GPS state of the Chaser at each LOS acquisition time  $t_i$  to perform: therefore, another source of noise will be represented by the GPS errors on the satellite's position and velocity. For the analysis, it will be assumed that modern, state-of-the-art GPS sensors and state determination techniques will be employed.

Not only the position, but also the attitude of the Chaser is essential for the proper

execution of the IROD algorithm. As described in Section 3.2, the LOS measurements are acquired in the Camera Frame, but the relative position vector has to be expressed in the RTN frame. This means that a rotation from the Camera Frame to the RTN frame has to be performed, and this makes the attitude of the Chaser a necessary information for performing the IROD task. Modern spacecraft usually employ sensors like star trackers for the attitude determination. Although very precise, these sensors are affected by some residual noise which is usually expressed in terms of angular deviation between the real and the measured orientation of the body axes of the spacecraft. Last, the satellites employed for rendezvous missions feature an active attitude control system, which often sees reaction wheels as the main attitude actuators. Thus, there will be discrepancies between the commanded and the actual control torques magnitude and direction, which have to be properly considered when assessing the IROD performance.

### 6.1.2 Performance assessment

To assess the IROD performance in the presence of noise, multiple simulations are conducted with different initial relative orbits. For each relative orbit, several simulations are performed, each with varying LOS errors, while the GNSS, star trackers, and attitude actuators remain unchanged. The tables below show the noise parameters that have been adopted:

Simulation #	Max LOS error [arcsec]
1	0
2	20
3	40
4	60
5	80
6	100

**Table 6.1:** Maximum LOS error for each simulation.

Body Axis	Mean [arcsec]	$\sigma$ [arcsec]
$\mathbf{X}_B$	0	5
$\mathbf{Y}_B$	0	5
$\mathbf{Z}_B$	0	30

**Table 6.2:** Star trackers errors mean and standard deviation.

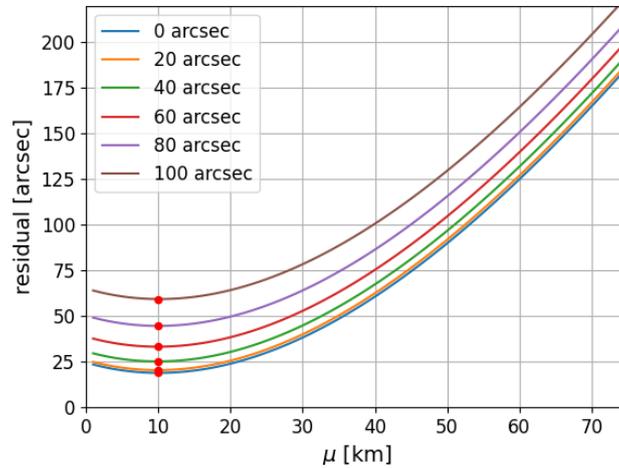
Component	Position Errors [mm]		Velocity Error [mm/s]	
	Mean	$\sigma$	Mean	$\sigma$
Radial	-2	103	0	0.5
Tangential	-7	282		
Normal	-15	197		

**Table 6.3:** GNSS error mean and standard deviation for position and velocity.

Module Error [N · mm]		Direction Error [arcsec]	
Mean	$\sigma$	Mean	$\sigma$
1	2.5	0	20

**Table 6.4:** Attitude actuators errors mean and standard deviation

The first investigated scenario involves an initial relative range of 10 *km* between Chaser and Target, on an orbit of approximately 500 *km* of altitude. The LOS acquisition accounts for the Target visibility along the trajectory, and a total of  $n = 1000$  measurements are acquired with an interval of  $\Delta t = 180$  *s* from one another. The values of the GNSS, star trackers and attitude actuators errors at each simulation step are drawn randomly according to the defined normal distribution; In contrast, LOS errors are sampled from a uniform distribution within a predefined range of  $[-max, +max]$ , where the maximum value is specified for each simulation in Table 6.1. What happens when the LOS error increases is shown in Figure 6.1: the residual curve  $m(\mu)$  is shifted upwards, but the abscissa of the minimum remains constantly the same, assuring that the IROD solution will not be affected much by the different noise parameters throughout the six simulations.



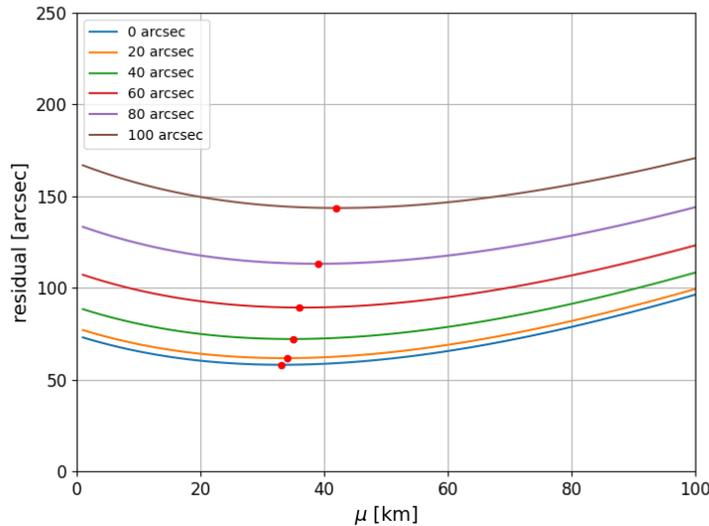
**Figure 6.1:** Residual curve with varying noise, 10 *km* of relative distance

The results of the relative orbit determination are reported in the table below. The code **IROD-XX** identifies the maximum LOS error featured in the simulation.

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	-3.27e-4	-38.5	10000	-2.34	240	-13.3	260
<b>IROD-00</b>	-3.45e-4	-36.4	9774	-1.77	230	-9.43	256
<b>IROD-20</b>	-3.46e-4	-36.2	9755	-1.79	230	-9.41	255
<b>IROD-40</b>	-3.47e-4	-36.0	9727	-1.81	229	-9.38	254
<b>IROD-60</b>	-3.49e-4	-35.6	9690	-1.82	228	-9.34	253
<b>IROD-80</b>	-3.50e-4	-35.0	9642	-1.83	227	-9.29	251
<b>IROD-100</b>	-3.52e-4	-34.4	9584	-1.84	235	-9.23	246

**Table 6.5:** IROD results with noise, inter-satellite distance of 10 km

Although degrading as the noise increases, the performance is still robust even at 100 arcsec, thanks to the precise determination of the correct scaling factor  $\hat{\mu}$ . The second set of simulations involves an initial relative range of approximately 30 km between the two spacecraft, and the IROD algorithm is executed following the acquisition of  $n = 2500$  LOS measurements spaced by 5 seconds from one another.



**Figure 6.2:** Residual curve with varying noise, 30 km of relative distance

As Figure 6.2 shows, the orbit determination here is more challenging due to the flatness of the residual curve, which makes it more difficult to accurately determine the scaling factor. Furthermore, it is possible to see how, unlike the previous case, the increasing noise tends to have a negative effect on the location of the minimum, which moves further away from the expected one as the noise increases. While this may be related to specific features of the case study which has been considered here, it may also be related to the fact that the same angular noise translates to a greater position error as the distance between the two object increases. For example, while 100 arcsec correspond to approximately 5 m at a distance of 10 km, for an

intersatellite distance of 30 km the associated position error would be of 15 m, degrading the orbit determination process. The detailed IROD results are reported in the table below.

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	-4.90e-4	-102	30383	-2.99	246	-2.75	249
<b>IROD-00</b>	-2.41e-4	-107	33119	-2.77	262	0.40	292
<b>IROD-20</b>	-2.13e-4	-108	34355	-2.93	271	0.47	302
<b>IROD-40</b>	-1.91e-4	-108	35274	-3.05	278	0.52	310
<b>IROD-60</b>	-1.83e-4	-108	35625	-3.13	281	0.51	313
<b>IROD-80</b>	-1.03e-4	-109	39184	-3.53	308	0.88	343
<b>IROD-100</b>	-4.24e-5	-108	42144	-3.87	330	1.20	368

**Table 6.6:** IROD results with noise, inter-satellite distance of 30 km

While these preliminary results constitute a good qualitative assessment of the performance in presence of noise, the overall behaviour under different mission scenarios is yet to be determined. Therefore, more in-depth analyses are conducted based on the Monte-Carlo simulations presented in the next Section.

## 6.2 Monte-Carlo analyses

### 6.2.1 Analyses set-up

As mentioned previously, the scope of the following Monte-Carlo analyses is to assess the overall performance of the algorithm under different environmental and operational conditions, as well as to determine an optimal number of measurements and an optimal frequency of acquisition that guarantee the highest accuracy in the relative orbit determination. To achieve this, several parameters must be randomized, with their values selected from a user-defined range. Specifically, the randomized features are the Chaser's absolute orbit, the relative orbit between Chaser and Target, the number of measurements and the time spacing between two consecutive measurements. Further details on the defined intervals for the selection of different parameters are provided in the Table below.

Chaser's KOEs	$a$ [km]	$e$ [-]	$i$ [deg]	$\Omega$ [deg]	$\omega$ [deg]	$u$ [deg]
	$R_{\oplus} + [400 \div 1500]$	$[10^{-7} \div 5 \cdot 10^{-3}]$	$[0 \div 110]$	$[0 \div 360]$	$[0 \div 360]$	$[0 \div 360]$
ROEs	$a\delta a$ [m]	$a\delta\lambda$ [km]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
	$[-150 \div 0]$	$[5 \div 75]$	$[-300 \div 300]$	$[-300 \div 300]$	$[-300 \div 300]$	$[-300 \div 300]$
Number of Measurements			$[500 \div 4500]$ with a step of 250			
Measurements time spacing			$[3 \div 120]$ s with non-regular progression			

**Table 6.7:** Monte Carlo setup parameters

Some intervals require further explanation and justification regarding how they are defined. For the Chaser's semi-major axis  $a$ , the orbit altitude is picked between 400 to 1500 km in accordance with the standard definition of LEO orbit. For the eccentricity  $e$ , the choice is dictated by the fact that the analytical motion model

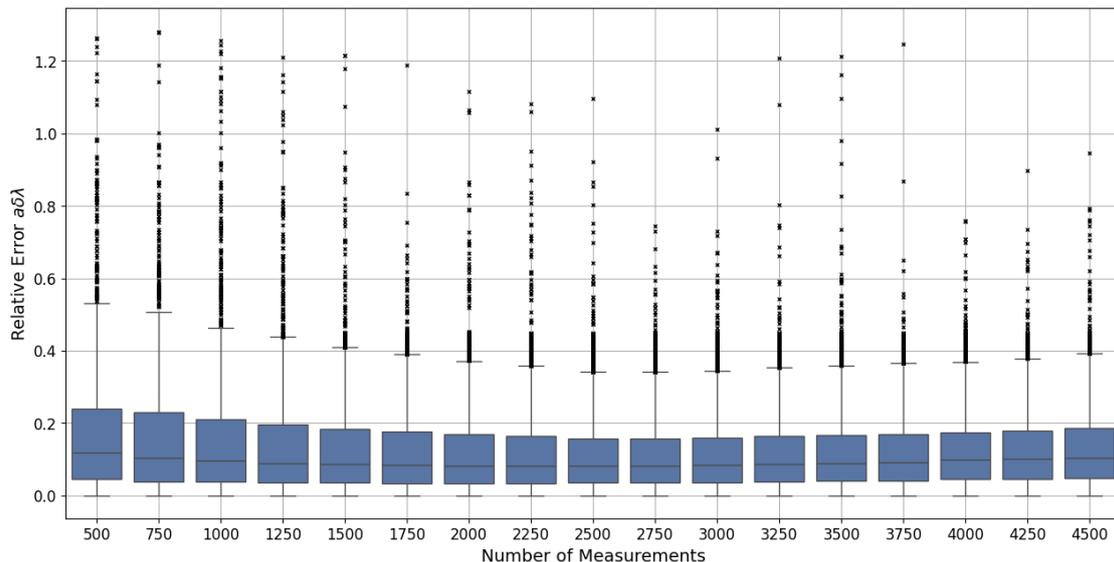
adopted for the IROD method is suited for quasi-circular orbits only. The inclination  $i$  is picked between  $0^\circ$  and  $110^\circ$  because it is very unusual to find spacecraft orbiting with higher inclinations, given the challenges posed in terms of required  $\Delta V$  for the launch.

Regarding the ROEs,  $a\delta a$  is chosen to be negative so that, in accordance with its definition, the Chaser's orbit will always be slightly higher than the Target's one. This is done to ensure that the Chaser drifts away from the Target without ever "overtaking" it, avoiding issues related to the sign of  $a\delta\lambda$  changing or to the complex camera pointing that would derive from that. Even though  $a\delta\lambda$  is picked between 5 to 75 km, the search space of the IROD algorithm is kept between 1 and 100 in order to increase the uncertainty of the problem.

### 6.2.2 Assessment of performance in function of measurements parameters

First, it is necessary to evaluate the behaviour of the IROD method in function of the number of measurements and of the measurements acquisition frequency, so that once optimal parameters for these quantities are found, the IROD accuracy can be measured under the best possible conditions.

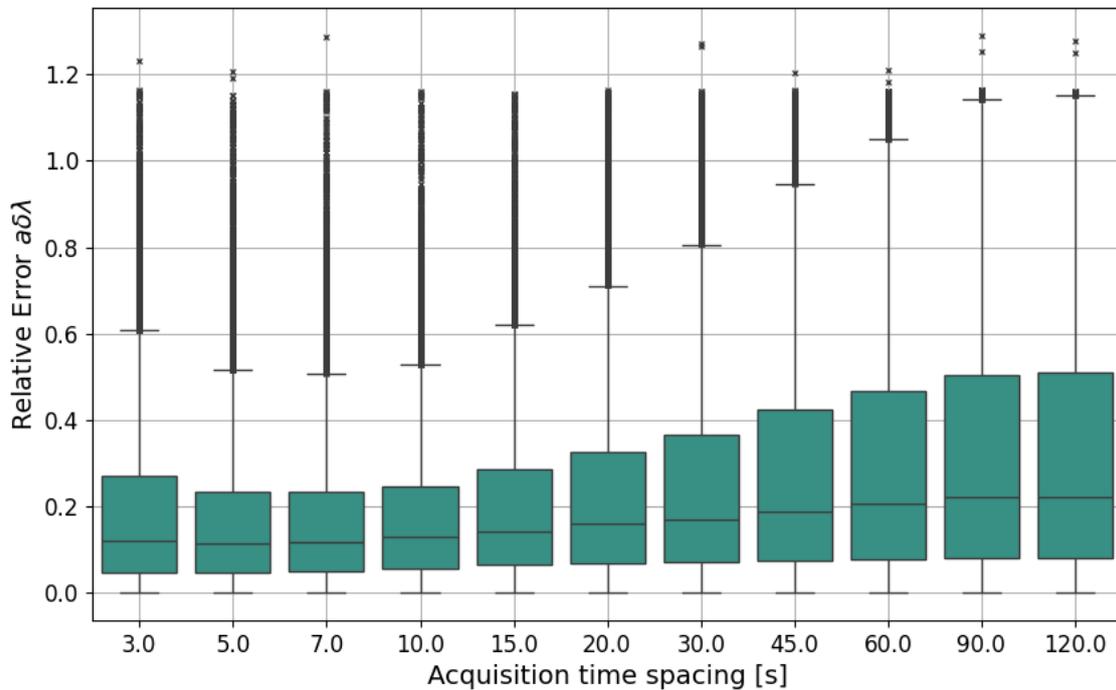
To do so, a total of approximately 130.000 simulations have been executed, each of them characterized by a random selection of the configuration parameters reported in Table 6.7. In this stage, the performance evaluation is based uniquely on the accuracy of the estimate of the relative range  $a\delta\lambda$ ; more detailed assessments will be conducted once optimal LOS acquisition parameters are determined. The Figure below shows the dependency of the relative error on the number of measurements:



**Figure 6.3:** Relative error of  $a\delta\lambda$  in function of the number of LOS measurements acquired

The correlation is somewhat weak, however it is possible to see a slight reduction of the relative error for a number of measurements between 2000 and 3500. This result may look counter-intuitive at first glance, since one would expect that the more

measurements are acquired, the higher is the accuracy of the IROD solution. However, it should also be acknowledged that acquiring more measurements increases on average the total acquisition time, which results in a higher divergence between the analytical propagation and the true trajectory at the end of the observation phase. For what concerns the frequency of acquisition, the results are shown in Figure 6.4.



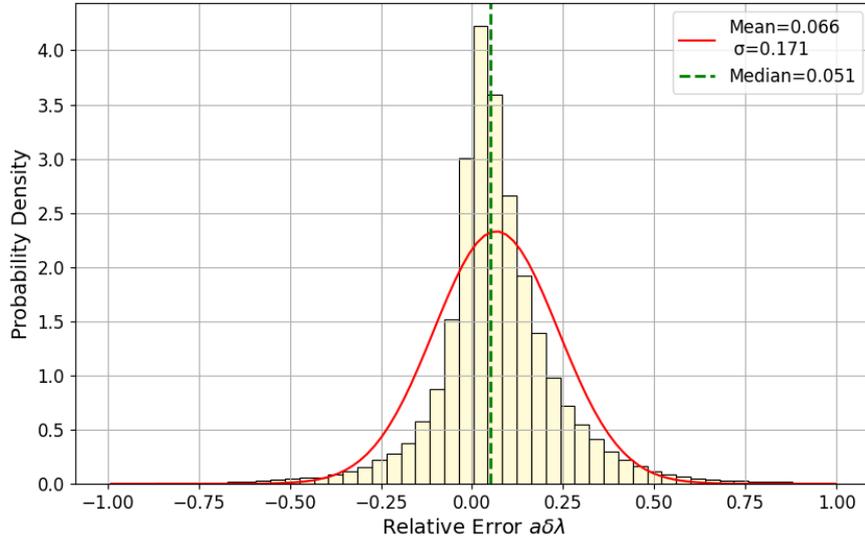
**Figure 6.4:** Relative error of  $a\delta\lambda$  in function of measurements acquisition interval

Here the correlation is stronger, and it is possible to note clearly how a measurements spacing of 5 to 10 seconds assures the highest accuracy. This result also highlights that, due to the need for a relatively short interval between consecutive measurements, the overall observation time will be somewhat reduced, or at least lower than it would have been with a longer time spacing. On a mission perspective, this implies that the Chaser will drift away from the Target of few kilometers at most, avoiding excessive increase of the  $\Delta V$  required to complete the rendezvous. Now that the measurements parameters have been at least restricted to a narrow interval, it is possible to assess the performance of the IROD method in terms of accuracy of the determined relative orbit.

### 6.2.3 Assessment of performance in terms of accuracy

The main focus of this section is to assess how closely the computed IROD solution matches the initial relative state, which is randomly determined at the start of each simulation. Since the measurements parameters have been already restricted to a relatively small interval, they will be picked from different, narrower ranges compared to the ones used before. In particular, for the number of measurements

the value will be picked from 2000 to 3250, while the time spacing will only assume the value 5, 7 or 10 seconds. This time, a total of approximately 100.000 simulations has been performed.



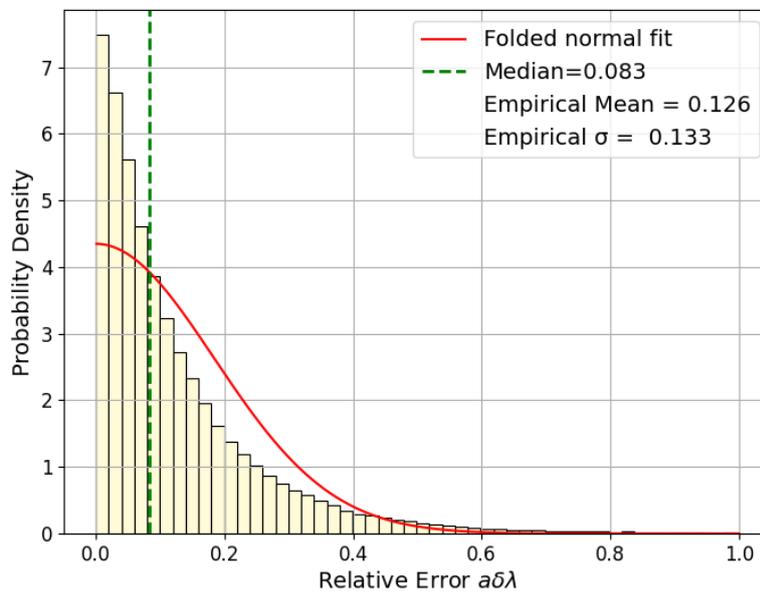
**Figure 6.5:** Distribution of  $a\delta\lambda$  signed relative error

Figure 6.5 shows how the relative error on the relative range follows a distribution which is not perfectly centered around 0, but it is rather shifted to the right towards the positive values. Given the formula of the signed relative error:

$$\xi = \frac{(a\delta\lambda)_{I\text{ROD}} - (a\delta\lambda)_{\text{actual}}}{|(a\delta\lambda)_{\text{actual}}|}, \quad (6.1)$$

this implies that the algorithm tends on average to slightly overestimate the along-track separation between Chaser and Target.

However, in order to properly evaluate the accuracy of the IROD method, the distribution of the relative error  $|\xi|$  should be considered instead:



**Figure 6.6:** Distribution of  $a\delta\lambda$  relative error

As it is also highlighted by the data on the plot, more than half of simulations results in an accuracy on  $a\delta\lambda$  which exceeds 90%. The data also shows that in the 80% of the simulations, the relative error falls under the threshold of 0.2, proving the reliability of the algorithm. In the following table, the performance is broken down accounting for the initial relative distance, to highlight possible differences in the accuracy in function of the initial relative range:

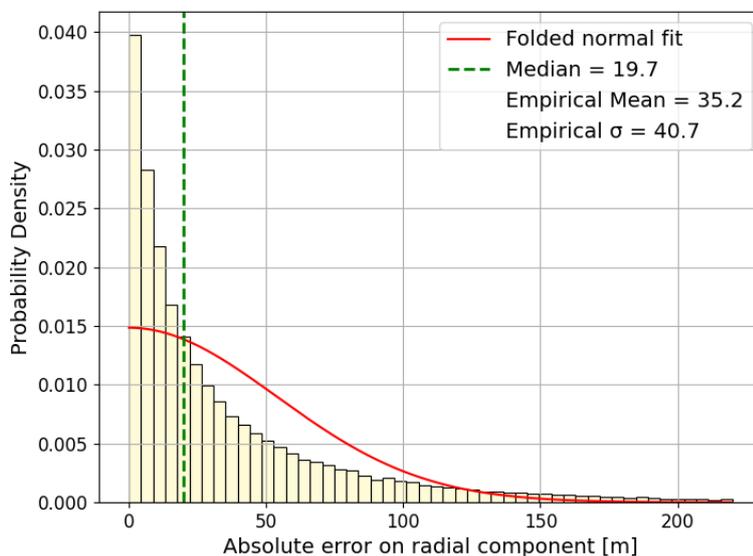
Interval [km]	Mean	$\sigma$	Median
5-15	0.187	0.175	0.134
15-25	0.121	0.121	0.085
25-35	0.103	0.113	0.068
35-45	0.104	0.119	0.066
45-55	0.116	0.136	0.072
55-65	0.125	0.130	0.082
65-75	0.127	0.109	0.094

**Table 6.8:** Summary of  $a\delta\lambda$  relative error in function of initial relative separation

It is possible to see that the best performance is achieved for intermediate initial along-track separations (25-55 km). It is likely that at lower distances, the performance decreases due to the reduced effect of the orbital path curvature, while at higher distances, the performance worsens due to the larger relative position errors introduced by the analytical motion models.

For what concerns the remaining ROEs, an evaluation based on the relative error would be somewhat ambiguous, since the distribution from which they are initially picked from is centered around zero. Errors of few meters could possibly translate into huge relative errors, misleading the performance assessment. Thus, the analysis focuses on the absolute errors in the radial and normal components of the relative position in RTN, which are primarily influenced by all the ROEs except  $a\delta\lambda$  and  $a\delta\dot{a}$ , whose influence is comparatively smaller. Defining  $\mathbf{r} = (r_R, r_T, r_N)$  the relative position vector in the RTN frame, the absolute error on the radial component is given by:

$$E_R = |(r_R)_{IROD} - (r_R)_{actual}|. \quad (6.2)$$



**Figure 6.7:** Distribution of absolute error on  $r_R$

The errors on the radial component of the relative position vector expressed in the RTN frame add up to a few tens meters on average. Going more in details, it is possible to assess also how this position error relates to the distance at which the Target is being observed and tracked by the camera, just like it has been done before for the relative error on  $a\delta\lambda$ .

Interval [km]	Mean [m]	$\sigma$ [m]	Median [m]
5-15	30.5	36.7	17.0
15-25	24.6	30.4	13.5
25-35	25.0	30.6	13.7
35-45	30.0	35.5	16.6
45-55	37.0	41.1	21.7
55-65	45.3	46.5	28.3
65-75	54.7	50.6	38.3

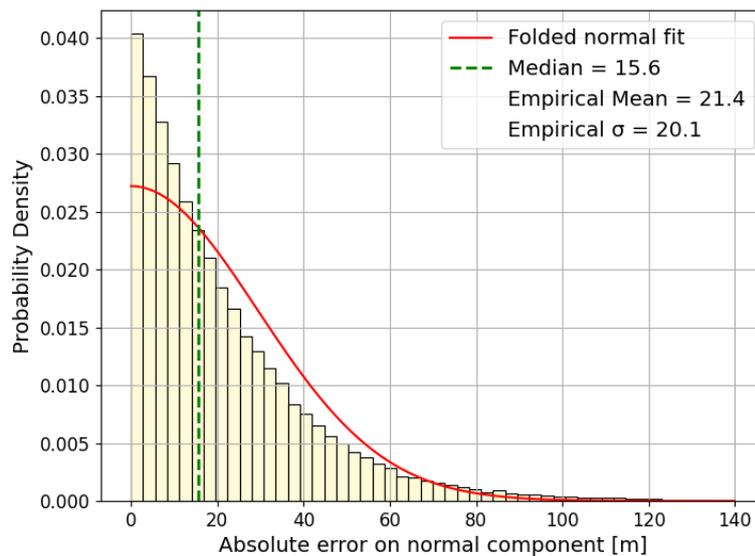
**Table 6.9:** Absolute error on radial component for different initial distance

Table 6.9 shows how once again the lowest errors are obtained for intermediate ranges (15-45 km) and how the error does not increase linearly as the initial relative distance gets bigger. The causes of this behaviour are very likely the same that were considered when analyzing  $a\delta\lambda$  relative error in function of the initial relative along-track separation.

Analogously at what has been done for the radial, the absolute error on normal component of the RTN relative position vector is given by

$$E_N = |(r_N)_{IROP} - (r_N)_{actual}|, \quad (6.3)$$

and it exhibits the following distribution:



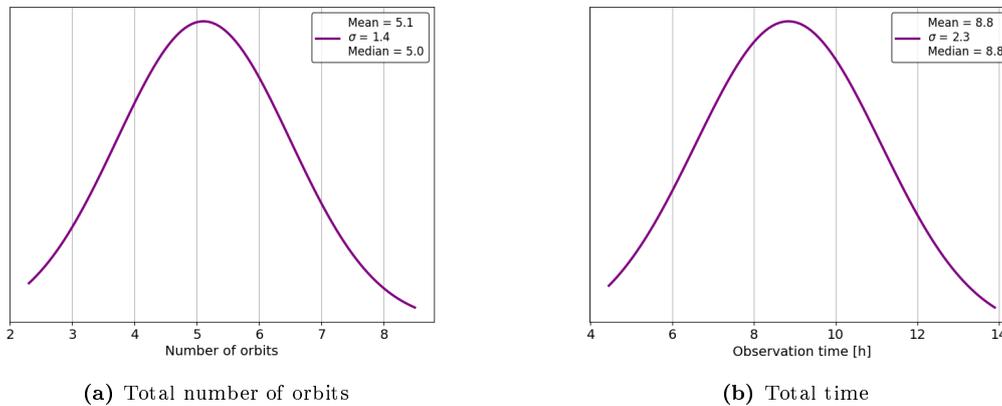
**Figure 6.8:** Distribution of absolute error on  $r_N$

Again, the errors add up to a few tens of meters like for the radial component. In relation to the initial relative range, the absolute error has the following behaviour:

Interval [km]	Mean [m]	$\sigma$ [m]	Median [m]
5-15	25.2	27.5	15.1
15-25	18.3	19.5	11.7
25-35	16.9	16.3	12.2
35-45	18.5	16.8	14.2
45-55	21.0	18.2	16.5
55-65	24.1	19.7	19.9
65-75	26.3	20.2	22.5

**Table 6.10:** Absolute error on normal component for different initial distance

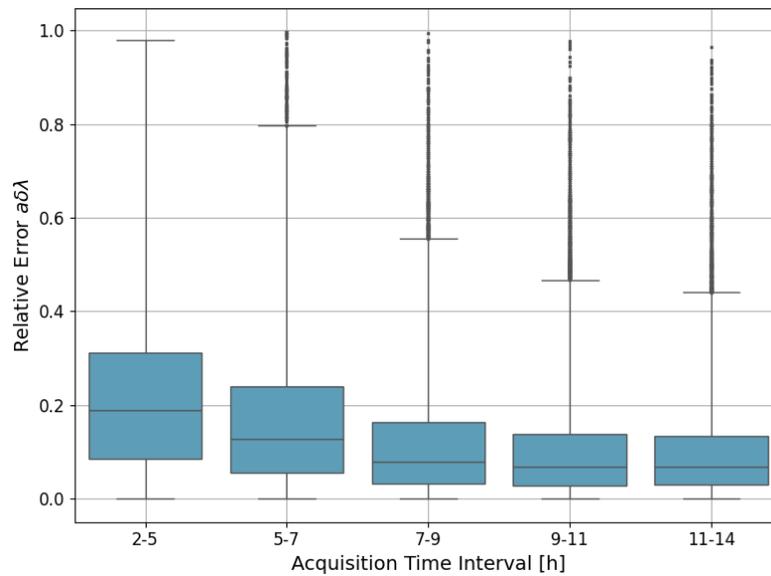
Table 6.10 proves that also for the normal component intermediate ranges are the ones characterized by the lower absolute error, for the same reasons explained above. A question which is yet to be answered however is how long the observation takes, and how this time relates to the accuracy. As it has been explained extensively in previous Chapters, the visibility constraint is always active and considered, implying that even if a measurement time spacing is defined, it may happen that no LOS measurements are actually acquired for minutes or even for hours. Since an optimization in terms of optimal trajectory, launch windows, orbit insertion and other *pre-rendezvous* mission phases is not the scope of this Thesis, the visibility time is not optimized or set as constraint, and it is therefore a consequence of the specific mission scenario which is randomly put together by the Monte-Carlo randomizer. The observation time is consequently evaluated, both in terms of number of orbits and in total hours of required observation. Results are shown below.



**Figure 6.9:** Durations of LOS acquisition process in terms of orbits and of time

The total observation time is aligned with the one experienced in previous similar missions and experiments, like PRISMA and AVANTI [16][50].

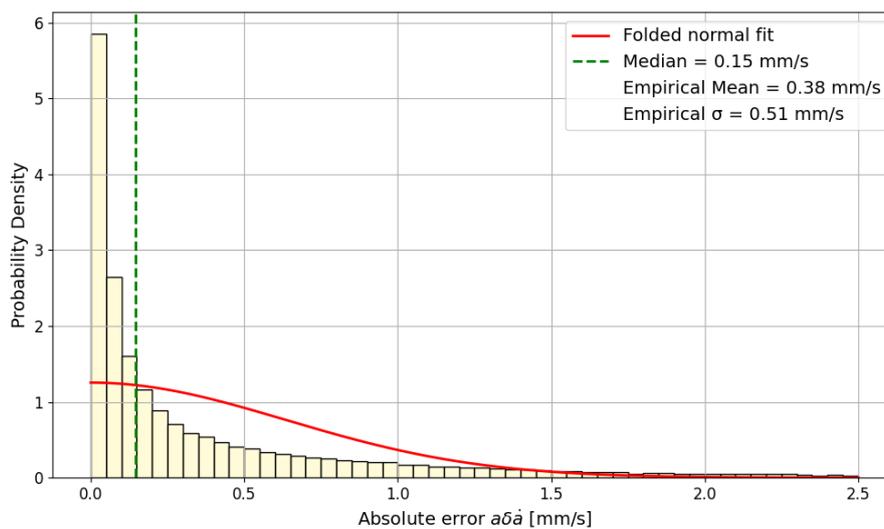
It is intended now to search for a possible correlation between the accuracy of the IROD solution and the total observation time. For conciseness, only the relative error on  $a\delta\lambda$  will be considered as performance index this time.



**Figure 6.10:** Accuracy in function of total observation time

As Figure 6.10 clearly shows, the accuracy increases remarkably as the observation time increases, although already achieving satisfying results for moderate acquisition times like 7-9 hours.

So far, of all the ROEs that compose the relative state vector, only the 6 related to the shape of the formation have been object of analysis, while  $a\delta\dot{a}$  is yet to be addressed and evaluated. As mentioned in Chapter 5, an accurate determination of  $a\delta\dot{a}$  is somewhat complicated, likely given the fact that, unlike the other ROEs, it is not directly related to the geometry of the relative trajectory and therefore not directly correlated with the LOS measurements. In detail, the performance in terms of absolute error is depicted in Figure 6.11:



**Figure 6.11:** Accuracy of  $a\delta\dot{a}$  determination

Even though an error of a few tenths of millimeters per second may seem negligible, it should be recalled that the true expected value of  $a\delta\dot{a}$  is on average in the order of  $10^{-5} \text{ mm/s}$ . Considering an unfavorable situation in which  $a\delta\dot{a}$  is determined with an error of  $1 \text{ mm/s}$  (which is quite unlikely given the distribution in Figure 6.11) and an observation time of 10 hours, this would determine an additional error of  $36 \text{ m}$  on  $a\delta a$  at the final time (*i.e.* the time at which the real time navigation begins) on top of the base error on the determination of  $a\delta a$  itself. This highlights that, although the overall performance of the IROD algorithm is robust and reliable, a more precise determination method of  $a\delta\dot{a}$  would be extremely beneficial for enhancing the global performance of the orbit determination process.

### 6.2.4 Performance assessment of the linear model solution

As mentioned in Chapter 5, if there exists a coarse estimation of the relative range (*e.g.* from TLEs), it may be possible to retrieve the initial relative state by computing only the solution of the linear motion model, appropriately scaled by the relative range itself. It is intended now to assess if this linear motion model solution is actually reliable in terms of accuracy, and how it performs compared to the full IROD algorithm. Naturally, the assessment cannot concern the accuracy on the determination of the relative range, since while for the full algorithm  $a\delta\lambda$  is a parameter to be estimated, for the linear solution it will be assumed to be known *a priori*. Therefore, the focus will be mainly directed towards the other ROEs of the relative state vector. For the analysis, the linear model solution will be computed accounting for a scaling factor  $a\delta\lambda$  to which, for each simulation, a random relative error between  $-5\%$  and  $+5\%$  with respect to the real relative range will be assigned. This choice is motivated considering the data processed during PRISMA and AVANTI, which highlighted a TLEs error in the along-track direction with a mean of roughly  $1 \text{ km}$  and a standard deviation of  $500 \text{ m}$ , at a distance of  $30 \text{ km}$  [51][52].

ROE	Absolute error - Full IROD			Absolute error - Linear solution		
	Mean [m]	$\sigma$ [m]	Median [m]	Mean [m]	$\sigma$ [m]	Median [m]
$a\delta a$	19.1	28.2	8.5	71.3	59.0	56.7
$a\delta i_x$	16.2	19.7	9.4	10.9	10.8	7.5
$a\delta i_y$	25.9	21.3	21.2	21.7	20.5	15.2
$a\delta e_x$	25.9	30.7	14.9	20.2	25.8	11.6
$a\delta e_y$	26.7	31.0	15.7	20.6	26.0	11.8
	Mean [mm/s]	$\sigma$ [mm/s]	Median [mm/s]	Mean [mm/s]	$\sigma$ [mm/s]	Median [mm/s]
$a\delta\dot{a}$	0.4	0.5	0.2	1.2	1.1	0.8

**Table 6.11:** Comparison of performance of full IROD method versus linear model solution in terms of ROEs absolute errors

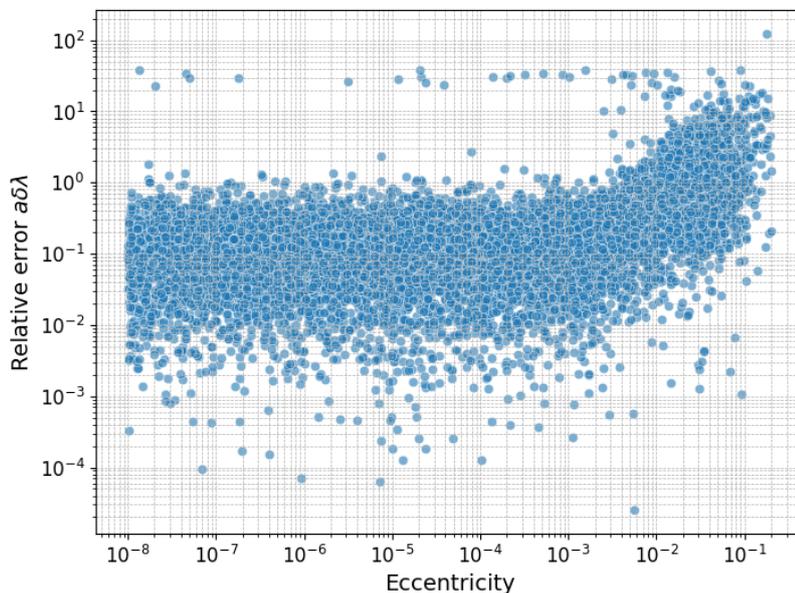
The results in Table 6.11 show how the full IROD method outperforms the linear model solution when it comes to determine  $a\delta a$  and its time derivative  $a\delta\dot{a}$ , likely due to the neglect of the curvature effects which are not considered in the linear model. On the other hand, the remaining ROEs computed with the linear model

solution are slightly more precise than the correspondent ones computed with the full IROD algorithm. The reader should not forget however that the linear model solution requires an *a priori* knowledge of the relative range (*i.e.* the scaling factor  $\mu$ ) to be computed: if leveraging the linear model solution would on one hand guarantee a slightly more accurate result, on the other hand it would decrease the autonomy of the GNC system and consequently of the entire rendezvous mission.

### 6.2.5 Performance assessment in function of eccentricity

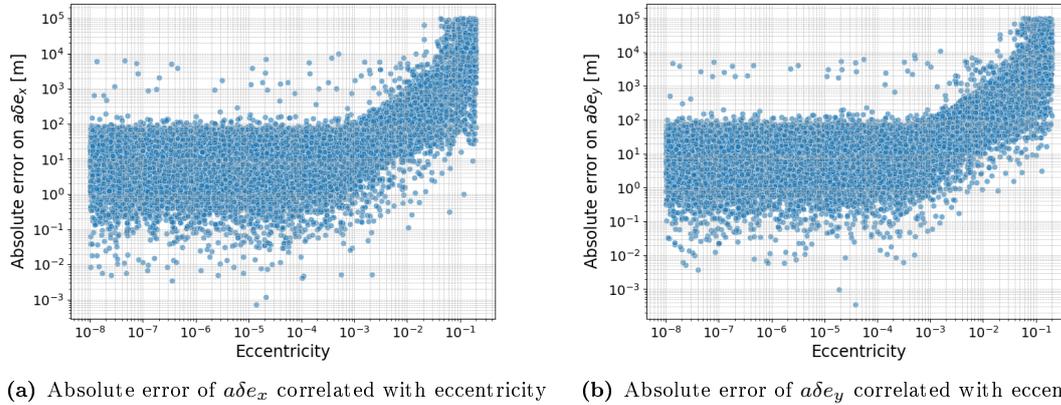
It is intended now to assess the performance of the IROD method with respect to the eccentricity of the Chaser's orbit. This evaluation does not concern the performance of the algorithm itself, but rather intends to provide guidance on the field of applicability of the IROD method. Previous Chapters have presented the analytical relative motion models as intended for circular or quasi-circular orbits ( $e \approx 0$ ). According to the Backbone Catalogue of Relational Debris Information (BACARDI) [53], more than 40% of LEO objects have an eccentricity lower than 0.001, and nearly 40% move on an orbit with eccentricity between 0.001 and 0.01. As explained thoroughly in [54], relative motion models like the HCW equations and especially the  $J_2$  and Drag ROE model used in the IROD algorithm perform very well for Chaser's eccentricities smaller than 0.01, while above this threshold the eccentricity effects dominate over those associated with perturbations, making necessary the adoption of other relative motion models like Tschauner-Hempel [25]. As mentioned previously in Section 6.2.1, the Monte-Carlo analyses have been performed considering Chaser's eccentricities between  $1 \cdot 10^{-7}$  and 0.005, in accordance with the applicability field of the relative motion model employed in the IROD algorithm. For this specific evaluation however, the eccentricity boundaries have been extended to  $1 \cdot 10^{-8}$  on one side, and to 0.2 on the other. In total, 50.000 simulations have been run approximately.

The first evaluation involves as usual the relative error on  $a\delta\lambda$ , this time with respect to the Chaser's eccentricity:



**Figure 6.12:** Relative error on  $a\delta\lambda$  with respect to the eccentricity of the Chaser's orbit

Figure 6.12 clearly highlights how, shortly before  $e = 0.01$ , the relative error starts growing rapidly, while being nearly constant before this threshold. Similar considerations can be made regarding the relative eccentricity vector components,  $a\delta e_x$  and  $a\delta e_y$ .



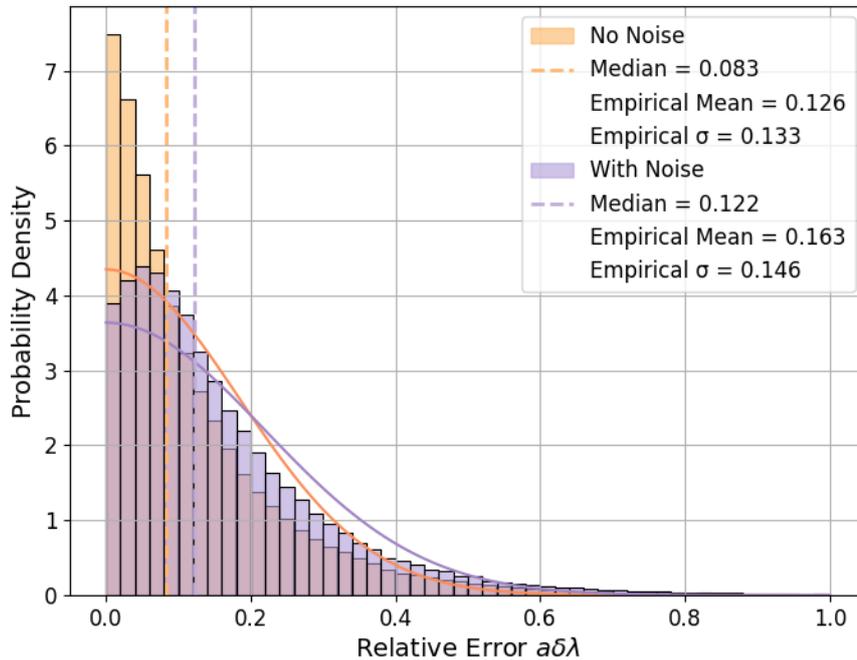
**Figure 6.13:** Error on relative eccentricity with respect to the eccentricity of the Chaser's orbit

These preliminary but already meaningful results limit the applicability of the IROD method to rendezvous missions involving spaceborne objects in circular or quasi-circular orbits ( $e < 0.005$ ). If one intends to extend the range of feasibility of the orbit determination algorithm, the adoption of a different relative motion model must be considered.

### 6.2.6 Performance assessment with respect to noise

The scope of this section is to assess the performance of the IROD method with respect to sensors and actuators noise and errors, extending the preliminary results that were shown in Section 6.1. To do so, the simulations are executed with the same exact input parameters that were used to evaluate the behaviour in the ideal case. This means that there is a one-to-one correspondance in terms of Chaser's KOEs, ROEs, measurements time spacing and number of measurements between correspondent variants of the Monte-Carlo simulations with and without noise. To explain this more clearly, the following example is proposed: since the simulations are run sequentially, each of them is associated with a variant number; in the Monte-Carlo with noise, simulation with variant number #1981 will have the exact same input parameters of simulation #1981 in the Monte-Carlo without noise. Obviously, the presence of noise and errors determines different results, even though the input parameters are the same. The noise and error parameters are kept the same as reported in Tables 6.2, 6.3 and 6.4 for star trackers, GNSS and attitude actuators, while the LOS error is set to have a maximum possible value of  $40''$ , in accordance with PRISMA and AVANTI's camera characteristics.

As done previously, the analysis first focuses on the performance of determining  $a\delta\lambda$ . Figure 6.14 shows how the accuracy is slightly degraded, when noise is taken into account. However, it can be safely said that the performance remains fairly robust, without massively diverging from the ideal case.



**Figure 6.14:** Noise effect on  $a\delta\lambda$  accuracy

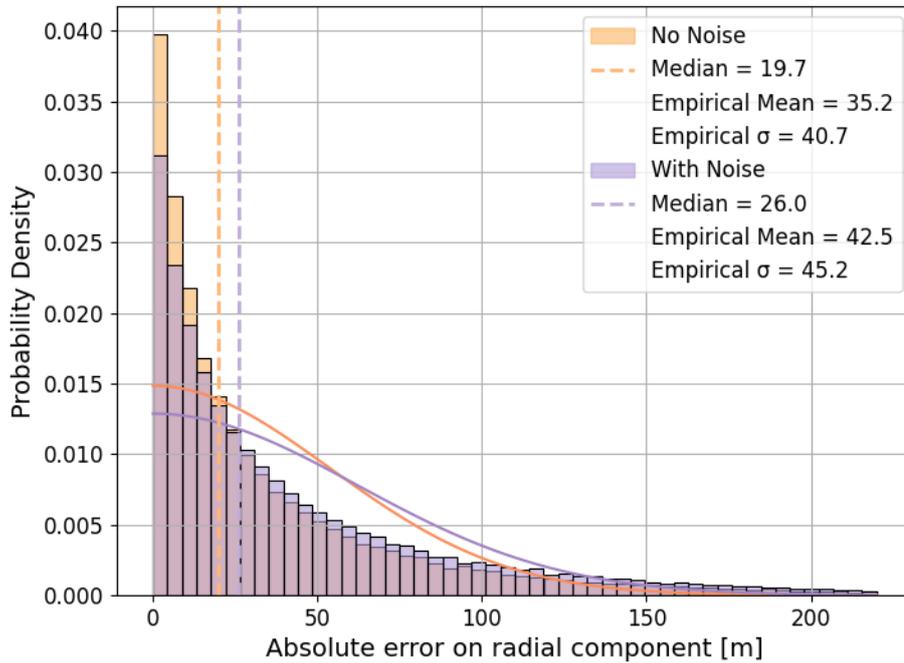
Analogously at what was done in Table 6.8, the relative error with respect to the initial relative distance follows the trends reported here:

Interval [km]	Mean	$\sigma$	Median
5-15	0.189	0.175	0.137
15-25	0.132	0.129	0.094
25-35	0.127	0.131	0.087
35-45	0.144	0.143	0.101
45-55	0.173	0.161	0.123
55-65	0.192	0.146	0.150
65-75	0.189	0.113	0.169

**Table 6.12:** Summary of  $a\delta\lambda$  relative error in function of initial relative separation, with noise

Unlike the case without noise and errors, it is possible to observe here how the error at further distances equals or even surpasses the error at closer ranges, likely due to the larger position error that the same angular error determines as the distance between the two object increases. If the best performances were obtained before for the range 25-55 *km*, the lower errors are now to be found at smaller distances, from 15 to 45 *km*.

Regarding the radial and normal components of the relative position vector, it is possible to compare the absolute errors with and without noise, analogously at what was done before (see Figures 6.7 and 6.8). Obviously, it is expected to have a degradation of the performance here as well, similarly to what can be observed for  $a\delta\lambda$ . Starting with the radial component:



**Figure 6.15:** Noise effect on radial component accuracy

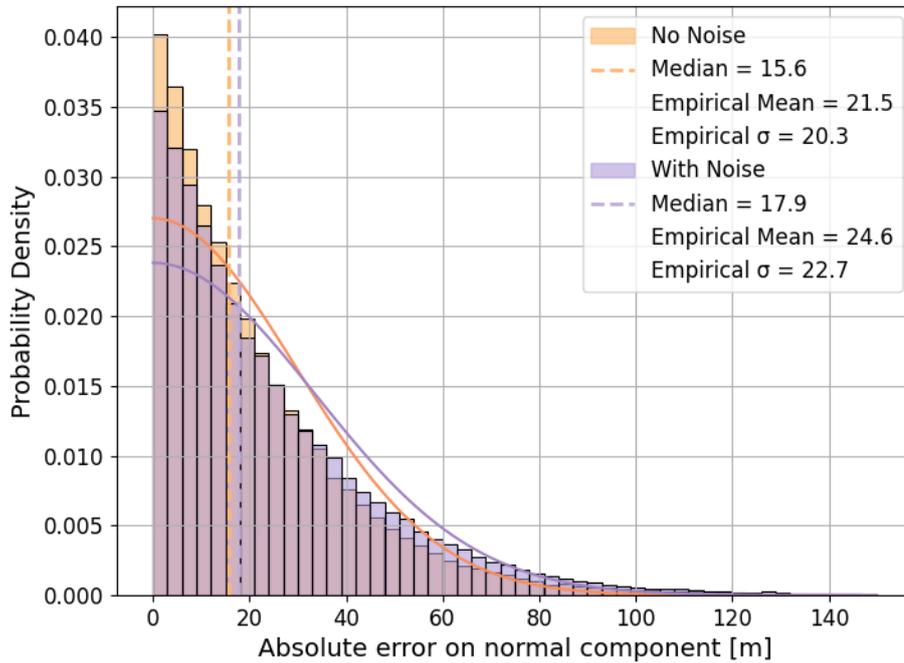
Regarding the accuracy with respect to the initial relative distance, it is reasonable to expect a behaviour similar to the one of the along-track separation, with finest accuracies moved towards closer initial tangential relative separations. Results are summarized in the Table below:

Interval [km]	Mean [m]	$\sigma$ [m]	Median [m]
5-15	30.9	37.2	17.4
15-25	26.1	31.4	14.5
25-35	28.6	33.7	16.3
35-45	36.4	39.9	22.0
45-55	46.8	45.7	31.0
55-65	59.4	51.3	44.0
65-75	71.6	52.9	59.9

**Table 6.13:** Absolute error on radial component for different initial distance, case with noise

Note the same behaviour as  $a\delta\lambda$ : accuracy at further distances decreases considerably, with respect to the case without noise (Table 6.9): if the best performance before was obtained for the range 15-45 km, the optimal is now shifted to shorter initial separations, from 5 to 35 km.

Last, the distribution of the absolute error on the normal component of the RTN relative position vector, shown in Figure 6.16:



**Figure 6.16:** Noise effect on normal component accuracy

Concerning the performance with respect to the initial relative distance, the following results are obtained:

Interval [km]	Mean [m]	$\sigma$ [m]	Median [m]
5-15	25.5	27.6	15.5
15-25	19.3	20.3	12.5
25-35	19.1	18.4	13.5
35-45	22.0	19.9	16.5
45-55	25.5	22.0	19.9
55-65	29.6	23.4	24.4
65-75	31.4	23.4	26.7

**Table 6.14:** Absolute error on normal component for different initial distance, case with noise

Again, one can note the degraded accuracy especially at larger initial along-track separations, with respect to the results in Table 6.10.

Given the results reported in this Section, it is possible to confirm that the IROD method shows overall robustness to the main sources of errors and noises that characterize its operational environment and conditions. As expected, the performance is slightly degraded with respect to the ideal case, but the extent of the deterioration is mild and contained, with generally small deviations from the noiseless results.

### 6.2.7 Estimation of initial covariance

As explained in Chapter 3, real-time navigation filters like Kalman Filters require not only an initial state vector (the IROD solution), but also an initial estimation of

the uncertainty on the state vector, expressed through the initial covariance matrix  $\mathbf{P}_0$ . The Monte-Carlo results can be effectively used for this scope: indeed, one can compute the variances of the ROEs evaluating the averaged absolute error from the estimated and expected ROE in each simulation. Being  $a\delta x$  a generic ROE, the variance is computed as:

$$\sigma_{a\delta x}^2 = \frac{1}{N} \sum_{i=1}^N ((a\delta x)_i^{I\text{ROD}} - (a\delta x)_i^{R\text{EAL}})^2, \quad (6.4)$$

where  $N$  is the total number of simulations. In other words, the variance is the square of the standard deviation  $\sigma$  of each ROE.

This approach considers the absolute error of the ROE whose variance is being computed. However, for  $a\delta\lambda$ , one would expect a larger absolute error and, consequently, greater uncertainty in terms of distance as  $a\delta\lambda$  increases. In other words, it would not be meaningful to use the same variance for a scenario where  $a\delta\lambda$  is 5 km compared to one where it is 75 km. Therefore, it is more appropriate to use the standard deviation of the relative error  $\sigma_{a\delta\lambda}^{rel}$  instead. Given an estimated value of  $a\delta\lambda$ , the variance can then be computed as follows:

$$\sigma_{a\delta\lambda}^2 = (\sigma_{a\delta\lambda}^{rel} \cdot a\delta\lambda)^2 \quad (6.5)$$

In order to consider the effects of noises and errors, the evaluation of  $\mathbf{P}_0$  will account for the Monte-Carlo results with noises.

ROE	$\sigma$	$\sigma^2$
$a\delta a$	42 m	1764 m <sup>2</sup>
$a\delta i_x$	30 m	900 m <sup>2</sup>
$a\delta i_y$	34 m	1156 m <sup>2</sup>
$a\delta e_x$	46 m	2116 m <sup>2</sup>
$a\delta e_y$	47 m	2209 m <sup>2</sup>
$a\delta\lambda$	0.12( $a\delta\lambda$ ) km	0.014( $a\delta\lambda$ ) <sup>2</sup> km <sup>2</sup>
$a\delta\dot{a}$	0.56 mm/s	0.31 (mm/s) <sup>2</sup>

**Table 6.15:** Standard deviation and variance for the ROEs.

The covariance matrix for the real-time navigation filter can be therefore written as:

$$\mathbf{P}_0 = \begin{pmatrix} \sigma_{a\delta\dot{a}}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{a\delta a}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{a\delta\lambda}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{a\delta i_x}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{a\delta i_y}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{a\delta e_x}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{a\delta e_y}^2 \end{pmatrix} \quad (6.6)$$

Further refinements could be made, for example adjusting the variance of the ROEs considering the initial estimated along-track separation. This would likely lead to

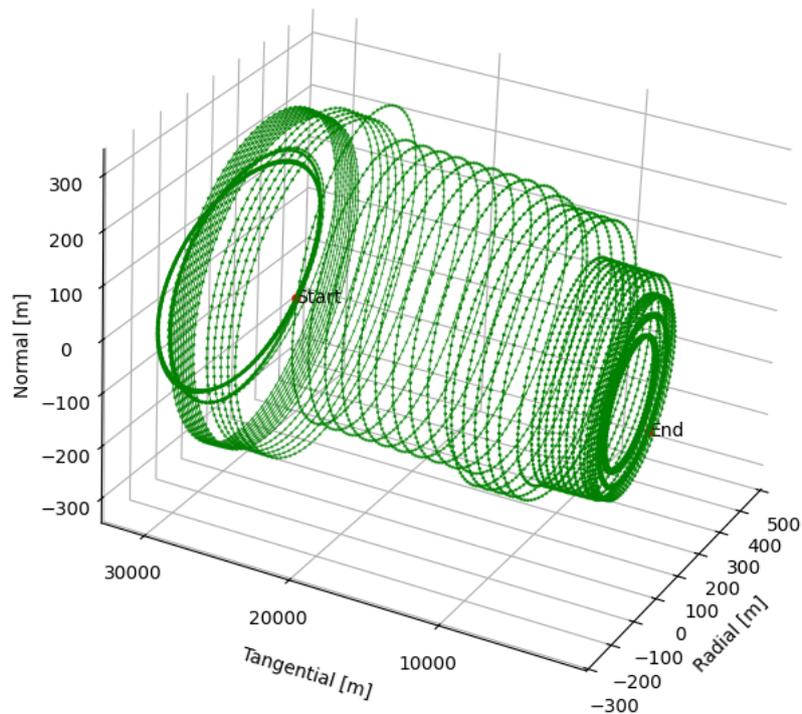
quicker convergence for intermediate distances, where it was observed that the errors on the radial and normal components reach their lowest values. However, this and other refinements will not be investigated throughout this Thesis.

## 6.3 Evaluation with ARGON data

This Section intends to provide a performance assessment of the IROD method which leverages the flight data of the PRISMA mission of SNSB and OHB-SE [55], in particular the one offering to the Advanced Rendezvous experiment using GPS and Optical Navigation (ARGON) experiment conducted by DLR [56]. First, an evaluation based on the GPS data of the experiment will be provided, performing the IROD with the LOS measurements extracted from the GPS information if Mango (Chaser) and Tango (Target). After that, a preliminary assessment of performance with image processing in-the-loop using real images from the mission will be presented.

### 6.3.1 Performance with GPS data

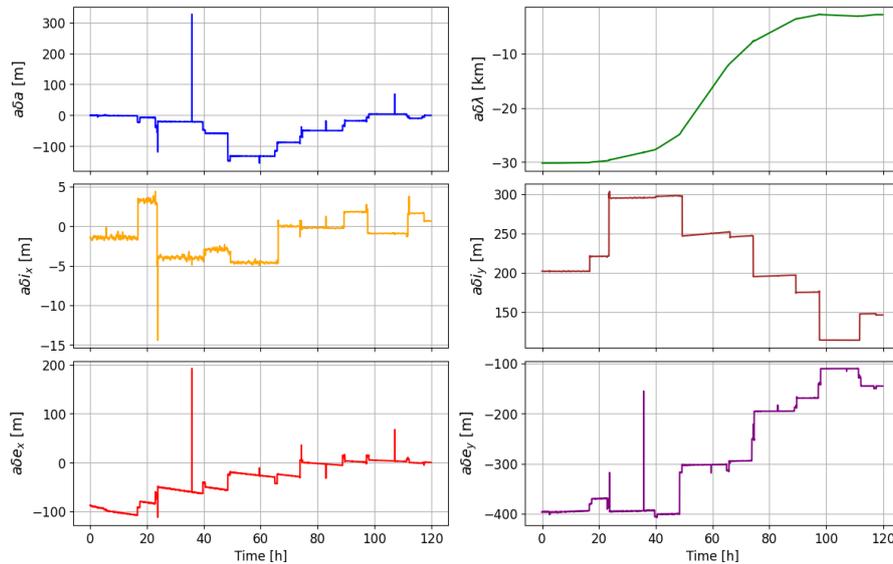
During ARGON, the spacecraft were initially separated by approximately 30 *km* in the along-track direction: over 6 days (22-27 April 2012), several manoeuvres were performed in order to rendezvous Mango with Tango, bringing the two object to the final relative distance of approximately 3 *km*.



**Figure 6.17:** Trajectory of Mango with respect to Tango during ARGON

The GPS data of Mango and Tango can be used to test the IROD algorithm along different relative orbit scenarios, by computing the LOS vector from Mango's camera to Tango and acquiring virtual LOS measurements which emulate those that would be computed after the Target's image acquisition and processing. This approach can be considered an hybrid between what was done so far and what one would have in a real angles-only rendezvous mission: while LOS measurements so far have been computed from the simulated spacecraft's positions, they are now extracted from the real data of the GPS receivers. What is still missing is the image acquisition and processing, which would naturally result in a further degradation of the LOS measurements accuracy and introduce further uncertainties and challenges for the orbit determination.

The IROD method is suited for the relative orbit determination in absence of manoeuvres: since during ARGON several manoeuvres were actually performed, it is necessary to filter out time windows where there is evidence of an unnatural change in the relative trajectory. To do so, it is possible to plot the ROEs of the formation, which clearly show how manoeuvres generate a step change in the ROE evolution.



**Figure 6.18:** ROEs evolution during ARGON

Having identified the timestamps and durations of manoeuvres, the IROD algorithm could be tested, acquiring for each run a number of measurements between 2000 and 3250, with a time spacing between two consecutive measurements randomly picked between 5 to 10 seconds. Few examples for different initial relative orbits are reported here:

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	1.50e-5	-20.5	-29475	-3.83	295	-51.0	-395
<b>IROD</b>	1.00e-4	-21.7	-28763	-1.99	293	14.2	-332

**Table 6.16:** IROD with ARGON GPS data, along-track distance of approximately 30 km

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	1.54e-5	-132	-22186	-4.42	247	-21.8	-303
<b>IROD</b>	3.92e-4	-120	-19234	-4.15	217	21.8	-233

**Table 6.17:** IROD with ARGON GPS data, along-track distance of approximately 22 km

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	1.48e-5	-49.3	-6454	-0.13	196	-1.60	-196
<b>IROD</b>	7.23e-4	-231	-8699	-1.71	249	18.8	-270

**Table 6.18:** IROD with ARGON GPS data, along-track distance of approximately 6 km

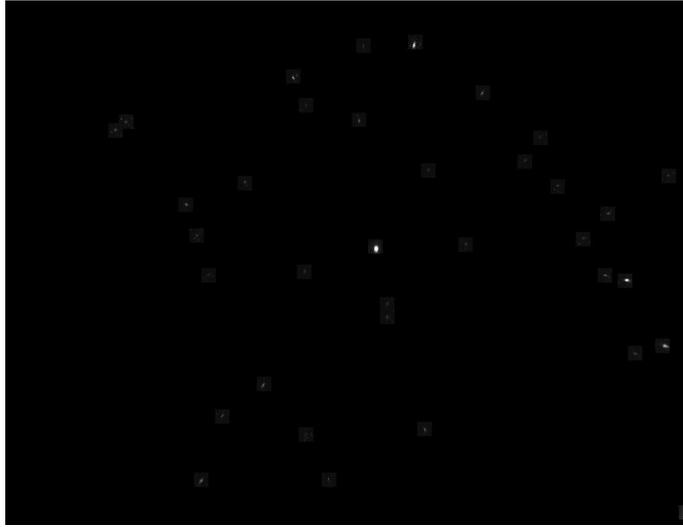
Note how the real initial  $a\delta\lambda$  is always negative, meaning that the approach was conducted along  $-V_{bar}$ . Thus, the initial search space of the binary search has to be adjusted accordingly, and is therefore set to  $[-100; -1]$  km instead of the preceding  $[1; 100]$  km.

Overall, the performance of the IROD method is not as robust as observed in the Monte-Carlo analyses: however, it is hard to assess if this is related to the method itself, whose performance may be affected by the different nature of the data employed, or if it is related to some specificities of the ARGON relative orbit, which may pose additional challenges to the state determination with this specific algorithm.

### 6.3.2 Preliminary performance assessment with ARGON images

So far, the LOS measurements have been virtually acquired from the simulated or from the real relative position vectors. However, it has been extensively discussed how in reality azimuth and elevation angles are the outputs of a complex image acquisition and processing task, which aims first at detecting the Target among all the bright objects in the Field of View and then tracks it along its motion, computing the LOS when required [34]. Since one of the main goals of ARGON was to demonstrate the feasibility of angles-only navigation for rendezvous with uncooperative Target, Mango was equipped with a camera tasked to acquire and process images of Tango, in order to enable LOS navigation in a Ground-In-the-Loop fashion [50]. The images captured during the experiment are still available, and could be used to test the implemented IROD algorithm in a scenario which is closer to reality than what was done so far.

Despite the 5-days duration of the experiment, only the images between 23rd and 24th of April 2012 are available: the dataset is therefore constituted by a total of roughly 1500 images covering a timespan of approximately 2 hours. The images are already partially processed, since the navigation sensor of Mango featured a process that autonomously extracts the so-called Region Of Interest (ROI) around the brightest objects in the FOV. Figure 6.19 shows one of the images downloaded during the experiment: among several bright objects which may include stars or different satellites, Tango can be identified with the bright spot in the centre of the image.



**Figure 6.19:** Image from the ARGON experiment. The bright spot in the centre is the Tango spacecraft

The images are then processed following a procedure which is very similar to the one described in Chapter 3, allowing the Target detection, the centroid's position calculation and the calculation of azimuth and elevation angles. These angles are converted into LOS vectors which are subsequently used to run the IROD algorithm. Unfortunately, the full IROD algorithm could not guarantee even a rough estimate of the relative orbit, failing in determining the relative range by approximately a factor 6. However, it cannot be concluded that the issue is solely due to having image processing in the loop, since the algorithm, when fed with GPS data over the same time window and with LOS measurements acquired at the same timestamps as the images, also resulted in failure. The most probable explanation is that, for reasons that are yet to be determined, the IROD method is somehow not suited for this specific time window of this specific orbit. Further analyses are required to investigate the causes of the failure in the orbit determination process.

Despite this, some encouraging results can be observed when analyzing the linear model solution. Two cases are investigated: one where the LOS measurements are solely acquired from the images, and another where GPS-based LOS measurements are added to the input. In this second case, the GPS measurements are acquired after the image-based measurements, corresponding to the immediately following time window. The linear model solution is then computed and appropriately scaled by the real initial  $a\delta\lambda$ . Results are reported below:

	$a\delta\dot{a}$ [m/s]	$a\delta a$ [m]	$a\delta\lambda$ [m]	$a\delta i_x$ [m]	$a\delta i_y$ [m]	$a\delta e_x$ [m]	$a\delta e_y$ [m]
<b>Reference</b>	1.50e-5	-7.13	-29776	3.20	220	-83.5	-369
<b>IROD - IMG only</b>	8.87e-4	-205	-29776	25.4	327	-0.70	-387
<b>IROD - IMG + GPS</b>	3.06e-4	-99.4	-29776	6.70	274	-2.51	-320

**Table 6.19:** Linear solution performance with LOS from ARGON images

While some ROEs still exhibit relatively large deviations in terms of absolute position errors, others are instead close to the reference initial state, especially when

GPS measurements are added to aid the orbit determination process. However, the accuracy is still not sufficient, and further investigation and analyses are needed to identify the causes behind this performance. Additional testing with images pertaining to other phases of the experiment would also be very beneficial to narrow down the pool of possible performance degrading factors.

## 6.4 Runtime evaluation on Zynq Board

As discussed in Chapter 5, the IROD method is required to be computationally light enough to be suited for on-board application. It is well known indeed that satellite onboard computers usually feature lower processing power and storage capability, compared to ordinary computer processors like the ones used in common laptops. This means that at least in terms of computation time, the data reported in Chapter 5 (Tables 5.1 and 5.2) is not enough to assess the onboard implementation feasibility. As a consequence, the C++ code is deployed and executed on the Zynq Board to allow Processor-In-the-Loop (PIL) testing and runtime performance assessment.

The Xilinx Zynq™7020, in this Thesis also referred as Zynq Board, is a fully scalable System on Chip (SoC) platform ideal for industrial Internet of Things (IoT) applications. Such devices have recently attracted attention for hybrid and reconfigurable onboard computing due to their relatively low cost and wide market availability [57][58]. The Zynq 7020 is also integrated in the ScOSA (Scalable On-board Computing for Space Avionics) onboard computer of DLR, and may soon be launched to orbit in the related space mission [59][60]. The Zynq Board features a dual-core ARM Cortex-A9 MPCore with a clock speed of up to 866 MHz. Previous experiments have shown that the runtime of other C++ programmes deployed on the Board was roughly 20 times slower than on standard desktop computers' processors belonging to the same family of the Intel Core i9-13950HX which was used in Chapter 5 for the preliminary assessment [58]. Therefore, a similar performance is to be expected also for the IROD algorithm.

The assessment is conducted evaluating the processing time for 4 different numbers of total measurements, and the computation time for the linear model solution is measured for both the SVD and the least-squares based approach.

Measurements	Linear Model Solution [ $\mu$ s]		Full IROD [s]
	SVD	Least-Squares	
250	2912	22	1.0
500	6538	24	3.9
1000	13702	23	15.6
2000	28088	23	60.3

**Table 6.20:** Runtime performance on Zynq Board

Compared to the data gathered in Table 5.2, it is possible to observe the expected increase of the runtime by approximately a factor 20. Also, the quadratic dependency of the computation time over the number of measurements is clear and evident,

allowing to estimate the runtime also for those number of measurements that were not tested here. Table 6.20 highlights the significant difference between the two proposed solvers for the linear model solution, confirming that the least-squares method is the preferred choice, even though the total runtime is primarily influenced by the other processes of the IROD algorithm. The total processing time of approximately 1 minute with 2000 measurements can be considered a symbolic but meaningful benchmark for future tests, refinements and optimizations.



# 7

## Conclusion

The goal of this Thesis was the presentation the work related to the implementation and testing of an algorithm for Initial Relative Orbit Determination through angles-only measurements. The IROD algorithm leverages a non-linear coordinate mapping that allows to overcome the intrinsic non-observability problem of Line-Of-Sight relative navigation for space rendezvous. Relying on a simple but effective binary search method, the algorithm retrieves an estimate of the inter-satellite distance between Chaser and Target, and subsequently computes a coarse initial state vector of the satellites formation. This coarse estimate, alongside with the associated covariance matrix, can be used to initialize a real-time navigation filter. The algorithm has proven to be robust to several sources of noise and errors typical of vision-based rendezvous missions, most notably the LOS error. The method has been tested through extensive Monte Carlo simulations, demonstrating its performance across various mission scenarios and ultimately confirming the overall accuracy of the computed solutions. The IROD method has also been tested using real data from the PRISMA mission, yielding promising results, though not yet conclusive enough to confirm its full reliability under the conditions of a real scenario. Finally, the algorithm has been executed on a platform that could serve as a satellite on-board computer in the near future, demonstrating both its suitability for onboard implementation.

Future work should primarily focus on integrating image processing into the loop. In this study, LOS measurements were obtained from the high-fidelity dynamic simulator, which inherently produced a perfect match with the true relative position unit vector. However, in a real camera-based rendezvous scenario, the LOS vector must be extracted from acquired images, a process that, even if precise, will never achieve the same level of accuracy as numerical trajectory propagation. For a hypothetical mission design process, using simulated images would likely suffice to evaluate the behavior of the IROD algorithm under conditions closely resembling reality. Another improvement could be represented by the adoption of a more accurate relative motion model, which would ultimately increase the accuracy of the computed IROD solution. As highlighted especially by the Monte-Carlo simulations, there is also the need for an alternative method to calculate the differential drag parameter, whose computation did not exhibit excellent performance overall. Moreover, in light of a potential future vision-based rendezvous mission, a system-level optimization of the code should be conducted, for example improving memory and resources man-

agement. Lastly, it is essential to evaluate how a real-time navigation filter, such as an Extended Kalman Filter, responds to the outputs of the IROD algorithm to determine whether its accuracy is sufficient for the intended purpose.



# 8

## Bibliography

- [1] J-S. Ardaens. *Angles-only relative navigation Low Earth Orbit*. PhD thesis, TU Delft, 2020.
- [2] K.T. Alfriend M. Willis and S. D’Amico. Second-order solution for relative motion on eccentric orbits in curvilinear coordinates. *arXiv preprint arXiv:1909.02146*, 2019.
- [3] W. Fehse. *Automated Rendezvous and Docking of Spacecraft*. Cambridge University Press, Cambridge, United Kingdom, 2003.
- [4] Alex Ellery, Joerg Kreisel, and Bernd Sommer. The case for robotic on-orbit servicing of spacecraft: Spacecraft reliability is a myth. *Acta Astronautica*, 63(5-6):632–648, 2008.
- [5] Joseph H Saleh, Elisabeth S Lamassoure, Daniel E Hastings, and Dava J Newman. Flexibility and the value of on-orbit servicing: New customer-centric perspective. *Journal of Spacecraft and Rockets*, 40(2):279–291, 2003.
- [6] Andrew M Long, Matthew G Richards, and Daniel E Hastings. On-orbit servicing: a new value proposition for satellite design and operation. *Journal of Spacecraft and Rockets*, 44(4):964–976, 2007.
- [7] Angel Flores-Abad, Ou Ma, Khanh Pham, and Steve Ulrich. A review of space robotics technologies for on-orbit servicing. *Progress in aerospace sciences*, 68:1–26, 2014.
- [8] Florian Rems, Eicke-Alexander Risse, and Heike Benninghoff. Rendezvous gnc-system for autonomous orbital servicing of uncooperative targets. 2017.
- [9] Per Bodin, Ron Noteborn, Robin Larsson, Thomas Karlsson, Simone D’Amico, Jean Sebastien Ardaens, Michel Delpech, and Jean-Claude Berges. The prisma formation flying demonstrator: Overview and conclusions from the nominal mission. *Advances in the Astronautical Sciences*, 144(2012):441–460, 2012.
- [10] G. Gaias and J-S. Ardaens. Flight demonstration of autonomous noncooperative rendezvous in low earth orbit. *Journal of Guidance, Control, and Dynamics*, 41(6), 2018.

- 
- [11] Gabriella Gaias, Jean-Sébastien Ardaens, and Christian Schultz. The avanti experiment: flight results. 2017.
- [12] J-S. Ardaens and G. Gaias. Angles-only relative orbit determination in low earth orbit. *Advances in Space Research*, 61(11), 2018.
- [13] S. D’Amico M. Willis. Fast angles-only relative navigation using polynomial dynamics. *Advances in Space Research*, 73(11), 2024.
- [14] Chenchao Dai, Hongfu Qiang, Degang Zhang, Shaolei Hu, and Baichun Gong. Relative orbit determination algorithm of space targets with passive observation. *Journal of Systems Engineering and Electronics*, 35(3), 2024.
- [15] S. D’Amico A. W. Koenig. Observability-aware numerical algorithm for angles-only initial relative orbit determination. In *2020 AAS/AIAA Astrodynamics Specialist Conference*, 2020.
- [16] J-S. Ardaens and G. Gaias. A numerical approach to the problem of angles-only initial relative orbit determination in low earth orbit. *Advances in Space Research*, 63(12), 2019.
- [17] David A Vallado. *Fundamentals of astrodynamics and applications*, volume 12. Springer Science & Business Media, 2001.
- [18] W. H. Clohessy and R. S. Wiltshire. Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, 1960.
- [19] Kyle Alfriend, Srinivas Rao Vadali, Pini Gurfil, Jonathan How, and Louis Breger. *Spacecraft formation flying: Dynamics, control and navigation*, volume 2. Elsevier, 2009.
- [20] S. D’Amico. *Autonomous Formation Flight in Low Earth Orbit*. PhD thesis, TU Delft, 2010.
- [21] S. D’Amico and O. Montenbruck. Proximity operations of formation flying spacecraft using an eccentricity/inclination vector separation. *Journal of Guidance, Control and Dynamics*, 23(3), 2006.
- [22] J-S. Ardaens O. Montenbruck G.Gaias. Model of  $j_2$  perturbed satellite relative motion with time-varying differential drag. *Celestial Mechanics and Dynamical Astronomy*, 123, 2015.
- [23] Joshua Sullivan, Sebastian Grimberg, and Simone D’Amico. Comprehensive survey and assessment of spacecraft relative motion dynamics models. *Journal of Guidance, Control, and Dynamics*, 40(8):1837–1859, 2017.
- [24] Koji Yamanaka and Finn Ankersen. New state transition matrix for relative motion on an arbitrary elliptical orbit. *Journal of guidance, control, and dynamics*, 25(1):60–66, 2002.

- 
- [25] J Tschauner and P Hempel. Rendezvous with a target orbiting in an elliptical orbit. *Astronautica Acta*, 11(2):104–+, 1965.
- [26] Hanspeter Schaub, Srinivas R Vadali, John L Junkins, and Kyle T Alfriend. Spacecraft formation flying control using mean orbit elements. *The Journal of the Astronautical Sciences*, 48:69–87, 2000.
- [27] Y. Yu B. F. Chao and C. H. Chung. Variation of earth’s oblateness  $j_2$  on interannual-to-decadal timescales. *Journal of Geophysical Research*, 125, 2020.
- [28] B. P. Graziano D. M. Prieto and P. C. E. Roberts. Spacecraft drag modelling. *Progress in Aerospace Science*, 64, 2014.
- [29] Christopher W T. Roscoe, Srinivas R Vadali, and Kyle T Alfriend. Third-body perturbation effects on satellite formations. *The Journal of the Astronautical Sciences*, 60:408–433, 2013.
- [30] Jay W McMahan and Daniel J Scheeres. New solar radiation pressure force model for navigation. *Journal of Guidance, Control, and Dynamics*, 33(5):1418–1428, 2010.
- [31] C. Colombo G. Gaias and M. Lara. Analytical framework for precise relative motion in low earth orbits. *Journal of Guidance, Control, and Dynamics*, 43(5), 2020.
- [32] David C Woffinden and David K Geller. Relative angles-only navigation and pose estimation for autonomous orbital rendezvous. *Journal of Guidance, Control, and Dynamics*, 30(5):1455–1469, 2007.
- [33] D. C. Woffinden and D. K. Geller. Observability criteria for angles-only navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3), 2009.
- [34] T. Boge H. Benninghoff, T. Tzschichholz and G. Gaias. A far range image processing method for autonomous tracking of an uncooperative target. In *12th Symposium on Advanced Space Technologies in Robotics and Automation*, 2013.
- [35] B. Braun and J. Barf. Image processing based horizon sensor for estimating the orientation of sounding rockets, launch vehicles and spacecraft. *CEAS Space Journal*, 15(3), 2023.
- [36] M. Wei T. Sun G.Wang, F. Xing and Z. You. Optimization method of star tracker orientation for sun-synchronous orbit based on space light distribution. *Applied Optics*, 56(15), 2017.
- [37] C. Colombo G. Borelli, G. Gaias. Rendezvous and proximity operations design of an active debris removal service to a large constellation fleet. *Acta Astronautica*, 205, 2023.
- [38] R.E. Kalman. A new approach to linear filtering and prediction problems. 1960.

- 
- [39] M. Khodarahmi and V. Maihami. A review on kalman filter models. *Archives of Computational Methods in Engineering*, 30(1), 2023.
- [40] M.I. Ribeiro. Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43(46), 2004.
- [41] D.K. Geller and A. Perez. Initial relative orbit determination for close-in proximity operations. *Journal of Guidance, Control, and Dynamics*, 38(9), 2015.
- [42] Keith A LeGrand, Kyle J DeMars, and Henry J Pernicka. Bearings-only initial relative orbit determination. *Journal of Guidance, Control, and Dynamics*, 38(9):1699–1713, 2015.
- [43] Mary T Stringer, Brett Newman, T Alan Lovell, and Ashraf Omran. Analysis of a new nonlinear solution of relative orbital motion. In *Proceedings of the 23rd International Symposium on Space Flight Dynamics*. Jet Propulsion Laboratory, 2012.
- [44] Oliver Montenbruck, Eberhard Gill, and FH Lutze. Satellite orbits: models, methods, and applications. *Appl. Mech. Rev.*, 55(2):B27–B28, 2002.
- [45] Martin Wermuth, Oliver Montenbruck, and Tom Van Helleputte. Gps high precision orbit determination software tools (ghost). 2010.
- [46] Gabriella Gaias, Jean-Sébastien Ardaens, Simone D’Amico, et al. The autonomous vision approach navigation and target identification (avanti) experiment: objectives and design. In *9th International ESA Conference on Guidance, Navigation & Control Systems, Porto, Portugal*, 2014.
- [47] Tom Sproewitz, Siebo Reershemius, Lars-Christian Hauer, Sebastian Fexer, Martin Schütze, and Birgit Suhr. Development, testing and in-orbit verification of a large cfrp helical antenna on the aisat mission. In *2020 IEEE Aerospace Conference*, pages 1–9. IEEE, 2020.
- [48] Jens Hauslage, Sebastian M Strauch, Olaf Eßmann, Ferdinand WM Haag, Peter Richter, Julia Krüger, Julia Stoltze, Ina Becker, Adeel Nasir, Gerhild Borneemann, et al. Eu: Cropis—“euglena gracilis: combined regenerative organic-food production in space”—a space experiment testing biological life support systems under lunar and martian gravity. *Microgravity Science and Technology*, 30:933–942, 2018.
- [49] John L Jørgensen, Troelz Denver, Maurizio Betto, Peter S Jørgensen, Hans-Peter Röser, Rainer Sandau, and Arnoldo Valenzuela. Microasc—a miniature star tracker. In *Small satellites for Earth observation, 4th international symposium of the international academy of astronautics*, pages 157–162, 2003.
- [50] Simone D’Amico, J-S Ardaens, Gabriella Gaias, Heike Benninghoff, Benjamin Schlepp, and JL Jørgensen. Noncooperative rendezvous using angles-only optical navigation: system design and flight results. *Journal of Guidance, Control, and Dynamics*, 36(6):1576–1595, 2013.

- 
- [51] Simone D'Amico, J-S Ardaens, and Robin Larsson. Spaceborne autonomous formation-flying experiment on the prisma mission. *Journal of Guidance, Control, and Dynamics*, 35(3):834–850, 2012.
- [52] Jean-Sébastien Ardaens, Gabriella Vittoria Gaias, et al. Angles-only relative orbit determination during the avanti experiment. In *26th International Symposium on Space Flight Dynamics (ISSFD)*, pages 1–8, 2017.
- [53] DLR. <https://bacardi.dlr.de/>, 2024.
- [54] Heike Frei, Léo Renaut, and Isabel Ibáñez Jiménez. Review and analysis of relative orbit models for spacecraft rendezvous. Manuscript under review, 2025.
- [55] Staffan Persson, Per Bodin, Eberhard Gill, Jon Harr, and John Jørgensen. Prisma-an autonomous formation flying mission. 2006.
- [56] Simone D'Amico, Jean-Sebastien Ardaens, G Gaias, Benjamin Schlepp, Heike Benninghoff, T Tzschichholz, T Karlsson, and JL Jørgensen. Flight demonstration of non-cooperative rendezvous using optical navigation. 2012.
- [57] Alan D. George and Christopher M. Wilson. Onboard processing with hybrid and reconfigurable computing on small satellites. *Proceedings of the IEEE*, 106(3):458–470, 2018.
- [58] Leo Renaut, Heike Frei, and Andreas Nuchter. Cnn-based pose estimation of a non-cooperative spacecraft with symmetries from lidar point clouds. *IEEE Transactions on Aerospace and Electronic Systems*, 2024.
- [59] Andreas Lund, Zain Alabedin Haj Hammadeh, Patrick Kenny, Vishav Vishav, Andrii Kovalov, Hannes Watolla, Andreas Gerndt, and Daniel Lüdtke. Scosa system software: the reliable and scalable middleware for a heterogeneous and distributed on-board computer architecture. *CEAS Space Journal*, 14(1):161–171, 2022.
- [60] Daniel Lüdtke, Thomas Firchau, Carlos Gonzalez Cortes, Andreas Lund, Ayush Mani Nepal, Mahmoud M Elbarrawy, Zain Haj Hammadeh, Jan-Gerd Meß, Patrick Kenny, Fiona Brömer, et al. Scosa on the way to orbit: reconfigurable high-performance computing for spacecraft. In *2023 IEEE Space Computing Conference (SCC)*, pages 34–44. IEEE, 2023.