POLITECNICO DI TORINO

Master's Degree in Aerospace Engineering

Master's Degree Thesis

Implementation of an Unscented Kalman Filter for Attitude Determination of a Satellite with a Rotating Antenna

Supervisors

Prof. Fabrizio Stesina

Dott. Francesco Manconi

Prof. Alessandro Battagliai

Candidate

Simone FONTANA

Academic Year 2024/2025

Abstract

As part of the development of ESA's Earth Explorer 11 mission, the Politecnico di Torino has contributed to the design of WIVERN (WInd VElocity Radar Nephoscope), a future Earth observation mission aiming at characterizing global winds, clouds and precipitation by employing a fast-rotating, conically scanning Doppler radar antenna. In this work the primary focus is on the line-of-sigh (LOS) Doppler relative velocity measurement errors, induced by the spacecraft Attitude Determination and Control System (ADCS). The estimation of the misalignment of the antenna pointing is critical, as errors in its determination directly affect the accuracy of the observed wind, as even small angles lead to relative velocity errors along the boresight due to the high orbital speed of the satellite. This study expands on previous work carried out at Politecnico di Torino, where during the past years a numerical simulator of the Attitude and Orbit Control System (AOCS) of the WIVERN multibody spacecraft was developed. Currently, the implemented determination system adopts a full suite of sensors (star trackers, gyroscope and encoder) and an Extended Kalman Filter (EKF) as the observer algorithm, delivering overall satisfactory results. However, for nonlinear estimation problems and highly complex systems, the EKF may not provide optimal performance, motivating the exploration of more advanced filtering techniques.

In this study, an Unscented Kalman Filter (UKF) is proposed for the WIVERN satellite attitude determination, exploring different solutions in its implementation. Initially, a standard UKF with a linear measurement model is considered. Subsequently, alternative approaches are investigated to further improve accuracy.

Different simulations in nominal conditions (i.e. with system parameters at expected without perturbations) and Monte Carlo analysis were performed, demonstrating that the attitude determination system can satisfy the maximum Absolute Knowledge Error (AKE) requirements of 100 μrad per axis and 1 m/s for the main body attitude and LOS velocity, respectively. Furthermore, the performances of the UKF and EKF were compared under similar conditions, to better assess the differences between the two solutions. In particular, the critical issues related to antenna unbalance were studied through simulations of worst cases and Monte Carlo campaigns, confirming that the implemented UKF presents acceptable performances and robustness.

Acknowledgements

For this thesis work, I would first like to thank my advisors, Fabrizio Stesina and co-advisor Alessandro Battaglia, who guided and supported me throughout the development of the project. I would like to express my sincere gratitude for the opportunity I was given, which allowed me to develop and deepen my skills in this field. The project, which lasted about a year, was a valuable learning experience during which both professors generously shared their knowledge with me, especially in the initial phase, helping to guide the work in the right direction.
A special acknowledgment goes to my co-advisor, Dr. Francesco Manconi, who assisted me throughout the entire thesis journey with great patience and availability. On several occasions during this process, I found myself disoriented, unsure of which direction to take or how to solve certain problems. Francesco consistently supported me with essential advice and guidance that made it possible to complete a thesis of this level. Furthermore, I would like to thank him especially on a personal level: I often had to make important decisions concerning my academic future, and even in those cases, he was able to guide me with competence and experience. I consider Dr. Manconi to be an extremely knowledgeable and capable person in this field certainly one of the most experienced individuals I have had the opportunity to meet. He was able to share with me knowledge that I had not acquired during my academic studies, significantly enriching my cultural and technical background. This thesis work has not only been a means to achieve a final goal, but also a true learning path that allowed me to gain fundamental insights into the development and design of complex systems. I wish him all the best for the future and I am confident he will continue his work with the same passion and dedication that he showed in supporting me during this final academic endeavor.
Finally, I would like to thank the Politecnico di Torino, which has always been a place of learning and growth throughout my academic journey not only in terms of knowledge in the field of Aerospace Engineering, but also as a space where I was able to grow as a person through the many experiences I lived. To me, the Politecnico has always felt like home: a place where I knew I could always find classmates, professors, and friends ready to support and help me along the way. I wish everyone the chance to embark on a journey of knowledge in a place like the Politecnico di Torino, because for me, it was truly an unforgettable experience.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the context of ESA's Earth Explorer 11, the agency aims to design a mission focused on Earth observation. This project is specifically dedicated to measuring wind within clouds and precipitation. Over the past few years, ESA has significantly increased its interest in Earth observation missions, following a similar trend seen among other space agencies worldwide. [1].
One of the two main candidates for this mission is WIVERN, short for WInd VElocity Radar Nephoscope. It is one of the two remaining candidates in the ESA Earth Explorer 11 call and is has currently finished phase A studies for selection at the 2025 EE11 User Consultation meeting [2] [3].
Insufficient data and knowledge about tropical cyclones is a major obstacle in weather forecasting, resulting in inaccurate predictions of extreme events. Tropical cyclones pose a significant threat, causing severe damage, flooding, and loss of life. The potential of the WIVERN mission lies in its ability to enhance our understanding of these phenomena through innovative radar technologies, improving both monitoring and forecasting capabilities [4].
Today, the growing threat of global climate change and the increasing vulnerability of communities to extreme weather events have created a strong need for more advanced atmospheric observation tools. According to the World Meteorological Organization (WMO), strong winds are the main driver of weather-related economic losses, with damages estimated at around 500 billion dollars [5].To improve the accuracy of numerical weather prediction (NWP) models and enhance early warning systems, it is crucial to gain a deeper understanding of wind dynamics, particularly within clouds, in this context. From this need arises the idea behind the WIVERN (Wind Velocity Radar Nephoscope) mission, proposed as a new frontier for observing three dimensional winds in the presence of clouds, as well as precipitation and cloud properties. Unlike earlier sensors like the Doppler lidar on the Aeolus satellite, which can only measure winds in clear skies or thin clouds WIVERN is designed to measure wind directly along the line of sight (LOS), even inside thick cloud systems like tropical cyclones, frontal systems, and convective bands [5] [6] [7].
The key element of the mission is the Doppler radar operating in the W-band (94 GHz), mounted on a satellite in low Earth polar orbit (500 km). This radar employs a new technology called Polarization-Diversity Pulse-Pair (PDPP), which estimates Doppler velocity by transmitting horizontally and vertically polarized pulses separated by 20 μs. A polarized pulse is a short electromagnetic signal (such as a light or radio impulse) with a defined polarization. This type of radar provides high temporal and spatial resolution

while maintaining the accuracy needed to derive wind velocity with an error of less than 2 m/s in the horizontal component.
Moreover, the elliptical antenna (2.9 × 1.8 m) rotates and performs a conical scan, tracing on each 5 second revolution a ground swath 800 km wide. This configuration allows for near global coverage with multiple passes at high latitudes, making WIVERN particularly suitable for studying polar regions and the oceans.
During its development, WIVERN drew on experience from previous missions such as CloudSat and EarthCARE, which provided valuable lessons about radar functionality and data processing models. However, WIVERN represents a qualitative leap: while CloudSat and EarthCARE were nadir pointing and provided only vertical profiles, WIVERN, thanks to its conical scanning, enables a much broader and three-dimensional spatial coverage, integrating data on wind, reflectivity, hydrometeor content, and cloud structure.
Thanks to its innovative technology, WIVERN would enable both the forecasting and understanding of complex meteorological phenomena. One of the most significant case studies involves tropical cyclones, for which the mission can provide 3D maps of horizontal winds in the troposphere, revealing regions of vertical shear, internal circulations, and the dynamics of frozen mass within the convective ring. Recent simulations based on WRF model data referring to Hurricane Milton (2024) demonstrate that WIVERN can monitor cyclone intensification in near real-time, estimating peak winds and observing the internal vertical structure with extremely high resolution [3].

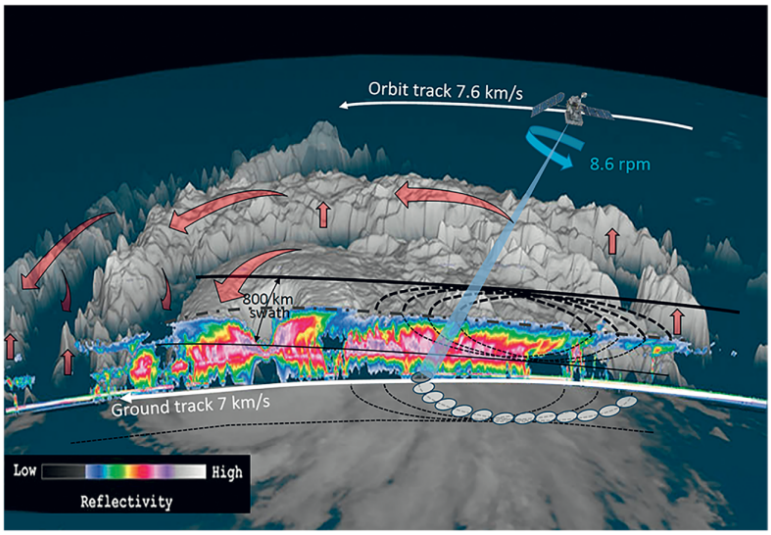

Figure 1.1: The WIVERN concept: the dashed black lines represent the path traced on the ground by the scanning beam. During each 7.5-second rotation, the satellite moves forward by 50 kilometres. The red arrows illustrate the general pattern of wind circulation.[credit [5]]

A Doppler radar used in meteorology is an instrument capable of detecting the presence and intensity of precipitation. This type of radar can measure the velocity of atmospheric particles (such as raindrops or snowflakes) along the radar's line of sight. This technology relies on the Doppler effect, which states that the frequency of the signal reflected by a moving target changes based on its relative velocity with respect to the observer.
The radar transmits radio pulses and receives the reflected echo. If the particles are moving toward the radar, the reflected frequency increases; if they are moving away, it

decreases. The frequency shift Δf is proportional to the radial velocity of the particle:

$$\mathbf{v}_r = \frac{\Delta f \cdot \lambda}{2} \tag{1.1}$$

where λ is the radar signal wavelength [8].
WIVERN uses a W-band radar (94 GHz) with a circular 3 meter antenna, mounted on a conically scanning platform. The radar measures the velocity component of winds and particles within clouds along the line of sight (LOS) using the Doppler shift of reflected radar signals. The satellite travels at approximately 8 km/s in low Earth orbit, generating a wide field of view.
WIVERN uses the W-band at 94 GHz for several reasons. First, the very short wavelengths (on the order of millimeters) typical of this band allow for high vertical resolution, which is essential for detailed analysis of cloud internal structures and detecting wind variations on kilometer scales. Moreover, this frequency is particularly sensitive to small particles, such as those found in thin clouds or ice-containing regions. Finally, another practical advantage is the possibility of using a compact antenna: with a 3-meter diameter, it is possible to generate a sufficiently narrow beam to ensure Doppler measurement accuracy without compromising the satellite's size or efficiency.
The radar rotates at 12 rpm with an off-nadir angle of 38°. This enables a conical scanning geometry, which allows for the measurement of the horizontal wind component. The satellite's motion and the radar beam width lead to a phenomenon known as "Doppler fading," or Doppler spectral broadening, which reduces the precision of velocity estimation. To counter this and improve resolution, polarization diversity is employed [9]. For a clearer overview of the mission's operation, refer to Figure 1.1 as presented in [5].
Traditional Doppler radars typically use a single polarization for transmitting and receiving signals. However, this approach leads to what is known as the "range–Doppler dilemma," where a balance must be struck between pulse repetition frequency (PRF) and the maximum unambiguous range. WIVERN overcomes this limitation by adopting a next-generation technique called polarization diversity (PD). In practice, the radar transmits two consecutive pulses with orthogonal polarizations horizontal (H) and vertical (V) separated by a short time interval. Each polarization is received on a separate channel, maintaining phase coherence between the two signals. This setup allows for a significant increase in pulse sampling frequency without introducing problematic ambiguities [9].
However, to ensure precise and reliable measurements, it is crucial to analyze and minimize observation errors. The principal sources of errors are pointing errors of rotating antenna, distortion on radar signal and errors inside the propagation model of radar wave [7] [10]. It is crucial for the antenna's elevation angle to be accurate at all times, as this correlates directly with the quality of the pointing. A small error in attitude determination, such as in satellite orientation, can propagate through to the radar's pointing direction, leading to significant measurement inaccuracies. In particular, this misalignment can cause errors in Doppler velocity measurements, which directly affect wind profile estimations. A pointing error distorts the observation angle, altering the measured Doppler velocity and thus the accuracy of the wind estimates [11].
To ensure high performance, a precision of the order of 100 μrad is required in both azimuth and elevation, along with high-precision antenna pointing. Specifically, the Absolute Performance Error on the antenna boresight should remain within $\pm 1000 \mu$rad. Additionally, accurate calibration of the pointing systems is essential for the proper operation of the system [11] [10].
First of all, it is necessary to develop a mathematical model to describe how deviations

in the antenna's orientation influence Doppler measurements. After that, reference data are created to estimate the systematic pointing error. By comparing the actual measurements with the estimated values, it becomes possible to correct deviations in the antenna's pointing direction. Subsequently, it is indispensable to use a filtering algorithm to compensate the effect on pointing error on measured Doppler velocity; this type of technology allows to reduce the measurement distortion. To improve antenna pointing accuracy, a useful method is the introduction of a Kalman Filter [10].
Filtering algorithms used in positioning navigation systems rely on both the non-linear state equation and the measurement equation. These algorithms operate in discrete time. In particular, the Kalman filter allows estimating the error or state of an object at step k based on measurements taken at the previous step $k-1$ [12]. In systems characterized by non-linear dynamics, as is the case for attitude, the equations governing object dynamics and observation are linearized by employing the extended Kalman filter (EKF). An alternative to the EKF is the Unscented Kalman filter (UKF). The UKF is a recursive estimation filter designed to better handle strongly non-linear systems with its inherent properties [12] [13].
The spacecraft model used in the WIVERN mission is a highly complex system due to the antenna's rotation on the body. This component generates various non-linear effects in combination with the spacecraft main body and Reaction Wheels (RWs). The intricate rotating components used in these missions might naturally display rotational imbalances, leading to internal forces and torques that impact the orientation of the spacecraft's platform [11]. This type of effect creates a linear kinematic model, but it involves a complex dynamic system influenced by external momentum, RWs, sensors and the primary effect, in particular, is the angular momentum generated by the rotating antenna.
The results obtained show that the use of the Extended Kalman Filter (EKF) is fully capable of meeting the mission requirements. However, it is important to highlight that the implementation of an Unscented Kalman Filter (UKF) could introduce significant improvements in terms of both accuracy and robustness, especially in complex applications involving multibody spacecraft with large rotating antennas. In such scenarios, the system dynamics are highly nonlinear and difficult to effectively model using the linear approximations on which the EKF relies. The UKF, on the other hand, adopts a more accurate nonlinear transformation strategy through the use of so-called sigma points, allowing for more precise estimates even in the presence of complex dynamics and non-Gaussian measurement noise.
The first attempt to mitigate the distortion effect is the implementation of an EKF filter, which introduces a linearization of the measurement model through the Jacobian of the measurement. The EKF implementation is similar to the linear Kalman filter, except that the Jacobian matrices must be computed at each time step to determine the locally linearized model of the system [14] [12]. Extended Kalman filter is easy to use and computably efficient. The results that were obtained with this type of filter are satisfactory. The time evolution of the Absolute Knowledge Error (AKE) of the line-of-sight velocity is a sinusoidal signal bounded between two limit values, -0.1 m/s and -0.7 m/s. Regarding the state-related AKE, the values are on the order of μrad. In particular, the Euler angles range from approximately 20 to 60 μrad in absolute value depending on the component. As for the satellite's angular velocities, they are centered around a mean value of 0 μrad/s for the angular rates around the z and y axes, while the rotation rate around the x axis is approximately 15 μrad/s. Finally, the AKE associated with the antenna rotation angle shows fluctuations ranging from -5 to $+5$ μrad [11].

For these reasons, this thesis aims to analyze in detail the potential advantages of using the UKF in such a context, with particular attention to the improvements in numerical results compared to those achieved with the EKF. During the development phase of the filter and its integration into the simulation environment, several key activities were carried out. First, it was necessary to define the dynamic model of the spacecraft, deriving the equations of motion that govern its behavior. This required an in-depth study of the state of the art to understand the structure and specific characteristics of the equations that accurately describe a multibody system with complex kinematic and dynamic constraints. This modeling phase is crucial for identifying and managing the main nonlinearities present in the system.
Next, the modeling of the sensors used to provide measurements to the filter was analyzed. This also required a careful review of the literature to understand the behavior and characteristics of each sensor, including star trackers, encoders, and gyroscopes. Each sensor model includes a detailed description of the measurement process and a noise model, aiming to replicate as closely as possible the different uncertainty contributions present in the acquired data.
Only after clearly and coherently defining both the system's dynamic model and the sensor measurement model was it possible to proceed with the design of the UKF. Unlike the EKF, this type of filter requires the introduction of a procedure for generating sigma points—points distributed around the current estimate in the state space—which are propagated through the system's non-linear equations. Additionally, it is necessary to define a measurement prediction model within the filter, capable of projecting these sigma points into the observation space in order to compare them with the real measurements and update the state estimate accordingly.
Finally, the algorithm was implemented and tested within the simulator, comparing the performance of the UKF with that of the EKF, in order to evaluate its effectiveness in terms of estimation error reduction, numerical stability, and the ability to track the dynamic changes of the observed system.
Subsequently, the new filter model can be integrated into the simulator through the development and coding of the corresponding algorithm. The simulator was built using Simulink, a graphical environment integrated into MATLAB that allows for modelling, simulating, and analysing multi-domain dynamic systems. Thanks to its block-diagram structure, Simulink facilitates the design and validation of complex algorithms, such as those for control, signal processing, and physical modelling. Therefore, in order to proceed with the implementation, it was necessary to thoroughly understand the functioning of the simulator in all its components, to ensure the correct integration of the filter within the existing architecture.
The first step in creating the model is to describe and identify the problems. It is necessary to develop the diagram that is useful for understanding the different parts of the problem and is also important for setting up the correct implementation. Essentially, the simulator is divided into four main blocks, each of which plays a specific role within the modelling and simulation of the system:

- The first block defines the mass and inertia properties of the system, including parameters such as total mass, center of mass, and inertia tensor. These physical properties are fundamental to determine the dynamic behaviour of the spacecraft.

- The second block implements the dynamic model, based on the equations of motion and integrated with the control model. This module allows the simulation of the

system's response to applied forces and torques, evaluating the effect of active control on the spacecraft's motion.

- The third block concerns the kinematics and includes the equations that describe the variation of attitude angles and body rotations. In particular, quaternions or Euler angles are used to represent the orientation of the body in space.

- Finally, the determination block contains both the sensor model and the estimation filter. This part simulates the measurements provided by onboard sensors (such as star trackers, gyroscopes, and encoders), and applies the Kalman filter to estimate the system state based on the observed data.

Once the overall simulator model was thoroughly understood, it was possible to proceed with the integration of the Unscented Kalman Filter (UKF) algorithm within the Determination block. This phase proved to be particularly challenging and required a significant time investment, as the implementation of the UKF presents several complexities, especially in managing the sigma points and correctly propagating them through nonlinear models. The main difficulties were related to the need to accurately interpret and translate the theoretical mathematical formulations into functional components within the simulator, while respecting the existing architectural specifications.
In particular, it was necessary to adapt the algorithm to the simulator's structure, ensuring consistency between the dynamic and kinematic models used and those required by the UKF. Only after ensuring the correctness of the implementation was it possible to proceed to the next phase.
The final phase of the work involved tuning the filter parameters, that is, the fine adjustment of some key variables such as the process noise covariance matrix, the measurement noise covariance matrix, and the weights associated with the sigma points. This process is essential to maximize the filter's performance in terms of accuracy, stability, and robustness. Several simulations were conducted with different configurations, analysing the impact of each parameter on the estimation error and comparing the results obtained with those of the EKF filter, in order to identify the most effective setup for the system under consideration.
The capability of the proposed solution to maintain reliable performance in the presence of system uncertainties has been assessed through Monte Carlo simulation campaigns. Additional analyses have been conducted to evaluate antenna unbalance, namely the effects of static and dynamic instabilities induced by the antenna.
The validation of the proposed strategy has been carried out using the data and requirements of the WIVERN mission. The scientific requirements of the mission include knowledge of the antenna orientation with an accuracy better than $100\,\mu$rad in both azimuth and elevation, as well as highly precise antenna pointing (i.e., an absolute boresight pointing error smaller than $1000\,\mu$rad). Furthermore, in terms of velocity along the boresight axis, the Absolute Knowledge Error (AKE) must be less than $1\,$m/s in absolute value.
A brief description of the thesis structure is provided. Chapter 2 presents the AOCS model of the system, both from a physical-mathematical perspective and in terms of its implementation within the simulator. Chapter 3 is dedicated to the determination system, describing the mathematical models of the sensors and their implementation in the Simulink environment. Subsequently, Chapter 4 illustrates the current EKF filter model and introduces the UKF filter to be implemented, highlighting the main differences between the two approaches. Chapter 5 details the implementation of the UKF filter in the

simulator, analysing the challenges encountered during the process. Chapter 6 discusses the results obtained from the simulations, with reference to the mission requirements and overall system performance. Finally, Chapter 7 contains the general conclusions of the work carried out.

Chapter 2

AOCS

This section describes the governing equations used to develop the spacecraft model and analyzes the implementation of the AOCS model in Simulink. The main focus is on formulating accurate dynamic and kinematic equations. To build a correct set of equations, it is essential to define a realistic and simplified spacecraft model.

2.1 Spacecraft model

The schematic representation of WIVERN spacecraft is made up of different parts. The simplified schematic of the WIVERN spacecraft is shown in Figure 2.1, composed of multiple rigid bodies, each identified by a number. These bodies maintain a constant shape over time relative to their local reference frames, although they are allowed to rotate around fixed axes.
The main structure of the spacecraft (body 1) is modeled as a hollow parallelepiped. Inside this structure, five reaction wheels (bodies 2–6) are fixed in a pyramidal configuration:

- wheels 2 to 5 are used for attitude control.
- wheel 6 neutralizes the antenna system's angular momentum.

To simulate disturbance effects, all wheels have static and dynamic imbalances. The point of connection between the rotating shaft with the parabolic reflector of antenna (body 8) and the main body is labeled O^A. A mass (body 7), shaped as a sphere, is positioned on the same shaft to compensate the imbalances introduced by the reflector's rotation.
There are different reference frames for for each body in the schematic model. The frame F_1 is placed at the center of mass of body 1 and is aligned with its principal axes of inertia. Each reaction wheel is associated with a specific reference frame,F_λ, where $\lambda \in \{2, 3, 4, 5, 6\}$. The reaction wheel frame F_λ is placed at the center of mass, with its $\hat{e}_z^\lambda$ axis aligned with the spin (rotation) axis of the wheel. For Body 7, the reference frame F_7 is placed at the center of the spherical body. Thanks to spherical symmetry, the orientation of F_7 can be arbitrarily chosen. The reference frame origin of F_8 is determined by its location at the apex of the parabolic reflector. The $\hat{e}_z^8$ axis is oriented along the central axis of the paraboloid. The frame F_A is rigidly connected to the antenna assembly A at the hinge point O_A, and its $\hat{e}_z^A$ axis is directed along the rotation axis of the supporting shaft[11].
Furthermore, it is necessary to define the external frame to describe the external force acting on the spacecraft. There are two reference systems:

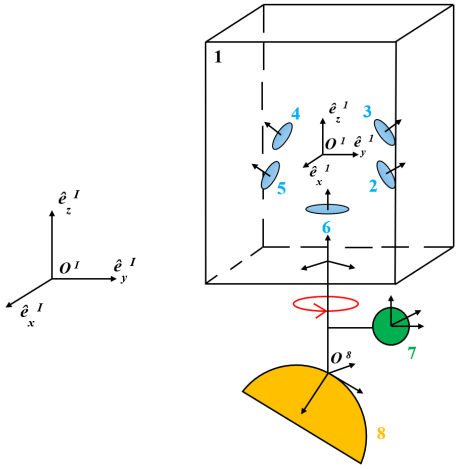

Figure 2.1: Spacecraft Model

- **ECI (Earth-Centered Inertial) — Frame F_I.**
- **LVLH (Local Vertical Local Horizontal) — Frame F_L.**
  The LVLH frame is a satellite-fixed reference system commonly used in orbital dynamics. Its $\hat{e}_{Lz}$ axis is aligned with the local vertical, pointing toward the center of the Earth, while $\hat{e}_{Lx}$ is directed along the orbital velocity vector. The $\hat{e}_{Ly}$ axis is oriented over the orbital angular momentum vector, completing a right-handed orthonormal frame [11].

2.2 Equation of motion

In this section, the governing equations that describe the motion of the spacecraft are presented. The two most important equations are the kinematic equation and the dynamic equations. These fundamental relations create the basis for modeling the attitude control of the spacecraft. The development of this part was carried out in previous work at Politecnico di Torino as part of Phase A activities [11].

2.2.1 Dynamic equations

The dynamics equations represent the balance of forces and torques on the satellite during its orbit.
The spacecraft includes imbalances that can create extra forces and torques.
The reaction wheel perturbation model consists of a small mass at the edge of the disk that causes a misalignment angle with the rotation axis. This misalignment shifts the center of mass of the reaction wheel. The model's perturbation values are based on the information provided by the reaction wheel manufacturer. The reaction wheel model is shown in Figure 2.2. To find the dynamics equations, we need to write the equations for the momentum and angular momentum for each part of the system. We start with the main body, called body 1. To properly describe the model, we use something called a "dyadic." A dyadic is a type of second-order tensor, written in a way that works well with vector math. Given the vector array representation of a generic basis:

$$\mathbf{e}_\lambda \triangleq \begin{bmatrix} \hat{e}_x^\lambda & \hat{e}_y^\lambda & \hat{e}_z^\lambda \end{bmatrix} \tag{2.1}$$

Vectors are represented as $\mathbf{r} = \hat{\mathbf{e}}_\lambda^\top \mathbf{r}_\lambda$, and dyadics as $\mathbf{E} = \hat{\mathbf{e}}_\lambda^\top \mathbf{I}_\lambda \hat{\mathbf{e}}_\lambda$. Both vectors and dyadics do not depend on the choice of reference frame, so it is not necessary to specify the coordinate system. The inertia dyadic of body λ measured about point O^i is represented by $\mathbf{I}_\lambda^i$.

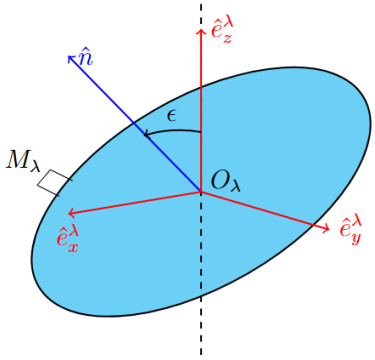

Figure 2.2: Reaction wheel Model

To write the correctly formulations, it is necessary to define the mixed inertia moments. These elements represent the moments respect to two different reference frames, and it is represent by $\mathbf{J}_{1/j}^\lambda$. Finally, to correctly define the moments acting on the body, it is necessary to define the distance (position vector) between two origins O^i and O^j, which is represented by $\mathbf{r_{i,j}}$.
To properly describe how the satellite moves, we need to write eight equations of motion. Then, we define the state vector output $\mathbf{y}$:

$$\bar{\mathbf{y}} = \begin{bmatrix} {}^1\bar{p}^T & {}^1\bar{h}_1 & {}_z^2\bar{h}_2^{(2)} & \dots & {}_z^6\bar{h}_6^{(6)} & {}_z^A\bar{h}_A^{(A)} \end{bmatrix} \tag{2.2}$$

This vector includes both the momentum and angular momentum balances of the main body $\mathbf{{}^1\bar{h}_1}$ and reaction wheels $\mathbf{{}_z^2\bar{h}_2^{(2)}}$.... It is also necessary to define the input vector:

$$\bar{\mathbf{x}} = \begin{bmatrix} {}^1\bar{v}_{I,1} & {}^1\bar{\omega}^1 & \Omega^2 & \dots & \Omega^6 & \Omega^A \end{bmatrix} \tag{2.3}$$

The input vector includes the translational velocity $\mathbf{{}^1\bar{v}_{I,1}}$, angular velocity of the main body $\mathbf{{}^1\bar{\omega}^1}$, angular velocity of the reaction wheels Ω^2, ..., Ω^6 and in the end the angular velocity of antenna $\mathbf{\Omega^A}$.
At this point, it is possible to write the equations in matrix form as:

$$\bar{\mathbf{y}} = \mathbf{A}\bar{\mathbf{x}} \tag{2.4}$$

Matrix $\mathbf{A}$ is made up of the following elements, which are the components of the equations that describe the model:

$$\mathbf{A} \triangleq \begin{bmatrix} m\mathbf{E} & -m\tilde{\boldsymbol{r}}_{1,\bullet} & -m_2\tilde{\boldsymbol{r}}_{2,②}\hat{e}_z^2 & \cdots & -m_6\tilde{\boldsymbol{r}}_{6,⑥}\hat{e}_z^6 & -m_A\tilde{\boldsymbol{r}}_{A,Ⓐ}\hat{e}_z^A \\ m\tilde{\boldsymbol{r}}_{1,\bullet} & \mathbf{I}_1 & \left(\mathbf{J}_{2/1}^{(2)} + \mathbf{I}_2^{(2)}\right)\hat{e}_z^2 & \cdots & \left(\mathbf{J}_{6/1}^{(6)} + \mathbf{I}_6^{(6)}\right)\hat{e}_z^6 & \left(\mathbf{J}_{A/1}^{(A)} + \mathbf{I}_A^{(A)}\right)\hat{e}_z^A \\ m_2\hat{e}_z^{2\top}\tilde{\boldsymbol{r}}_{2,②} & \hat{e}_z^{2\top}\left(\mathbf{J}_{1/2}^{(2)} + \mathbf{I}_2^{(2)}\right) & \hat{e}_z^{2\top}\mathbf{I}_2^{(2)}\hat{e}_z^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & 0 \\ m_6\hat{e}_z^{6\top}\tilde{\boldsymbol{r}}_{6,⑥} & \boldsymbol{e}_z^{6\top}\left(\mathbf{J}_{1/6}^{(6)} + \mathbf{I}_6^{(6)}\right) & 0 & \cdots & \hat{e}_z^{6\top}\mathbf{I}_6^{(6)}\hat{e}_z^6 & 0 \\ m_A\hat{e}_z^{A\top}\tilde{\boldsymbol{r}}_{A,Ⓐ} & \hat{e}_z^{A\top}\left(\mathbf{J}_{1/A}^{(A)} + \mathbf{I}_A^{(A)}\right) & 0 & 0 & 0 & \hat{e}_z^{A\top}\mathbf{I}_A^{(A)}\hat{e}_z^A \end{bmatrix}$$

At this point, the time derivatives of the linear and angular momenta are carried out for both the main body and the reaction wheels [11]:

$$\dot{\mathbf{p}} = -\tilde{\boldsymbol{\omega}}_1 \cdot \mathbf{p} + \mathbf{f} \tag{2.5}$$

$$\dot{\mathbf{h}}_1 = -\tilde{\boldsymbol{\omega}}_1 \cdot \mathbf{h}_1 - \tilde{\mathbf{v}}_{I,1} \cdot \mathbf{p} + \boldsymbol{\tau}_1 \tag{2.6}$$

$$\dot{\mathbf{h}}^{(\lambda)}_{\lambda,\text{rot}} = \hat{\mathbf{e}}^{(\lambda)}_z \cdot \left(-\tilde{\boldsymbol{\omega}}_1 \cdot \mathbf{h}^{(\lambda)}_\lambda - \tilde{\mathbf{v}}_{I,\lambda} \cdot \mathbf{p}^{(\lambda)} + \boldsymbol{\tau}^{(\lambda)}_\lambda\right) \tag{2.7}$$

2.2.2 Kinematic equation

The kinematic equations describe the evolution of a spacecraft's attitude and position over time as a function of angular and translational velocities, respectively. Kinematics equations are fundamental for determining the spacecraft's orientation and position during operations such as attitude control, navigation, and pointing manoeuvrers .
Regarding attitude, it's possible to represent this type of equations in two different ways: either using Euler angles or quaternions. The quaternion is a mathematical tool used to represent a rotation in three-dimensional space, without encountering singularity issues. For this reason, the quaternion representation is adopted. In the following chapters, the Euler angles representation will also be used to identify possible issues related to the Kalman filter implementation [15].
The equations governing the attitude kinematics of a satellite describe rotations about the **x**-axis. Integrating these equations allows to determine the satellite's attitude and relative orientation to the reference frame F_I. The relative orientation of each reaction wheel λ relative to the reference frame F_1.
The equations are given by:

$$\dot{q}^{I,1} = \frac{1}{2}\mathbf{B}\left(q^{I,1}\right)\ {}^1\boldsymbol{\omega}^1 \tag{2.8}$$

$$\dot{q}^{1,\lambda} = \frac{1}{2}\mathbf{B}\left(q^{1,\lambda}\right)\ {}^\lambda\boldsymbol{\Omega}^\lambda \tag{2.9}$$

The different elements represents:

- $(q^{I,1})^1$: quaternion in the reference frame F_I of body 1;
- ω^1: angular velocity body 1;
- $(q^{1,\lambda})^\lambda$: quaternion in the reference frame F_1 of body λ;
- Ω^λ: angular velocity for every Reaction wheel λ.

The matrix $\mathbf{B}$ is the quaternion kinetic matrix, which establishes the relationship between the quaternion derivative and the angular velocity.

$$\mathbf{B}(q) = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \tag{2.10}$$

The translational motion of the satellite's center of mass is represented by the following equation:

$$\dot{\mathbf{r}}_{I,\bullet} = \mathbf{v}_{I,\bullet} -^{1} \boldsymbol{\omega} \times \mathbf{r}_{I,\bullet} \tag{2.11}$$

The velocity of body 1 in an inertial reference frame is given by the absolute velocity of its center of mass in the inertial frame F_I, minus the relative velocity to the rotation of body 1 in the body frame F_1.

$$\mathbf{v}_{I,\bullet} = \mathbf{v}_{I,1} -^{1} \boldsymbol{\omega} \times \mathbf{r}_{I,\bullet} \tag{2.12}$$

To obtain the value of $\mathbf{v}_{I,1}$, it is necessary to know the translational momentum. The following momentum is known and has been derived from the equations of motion:

$$\mathbf{v}_{I,\bullet} = \frac{\mathbf{p}}{m} \tag{2.13}$$

Where $\mathbf{p}$ is translation momentum and $\mathbf{m}$ is mass of spacecraft [11].

2.2.3 Sliding mode control

A first-order Sliding Mode Control (SMC) law is used to keep the spacecraft pointing at nadir with an absolute antenna rotation speed of 12 rpm. The control torques generated by the SMC are sent to the reaction wheels to control the main body and the antenna. The antenna motor torque is incorporated into the overall control effort of the reaction wheels because it affects the main body.
The sliding surface, i.e. the condition or set of states in which the system is desired to remain, are defined as follows:

$$\mathbf{s} \triangleq \delta\mathbf{z} + \mathbf{K}_2 \delta\boldsymbol{\omega} \tag{2.14}$$

where $\mathbf{s} \in \mathbb{R}^4$ and $\delta\mathbf{z}$ represents a differential variation, the error on the control state, as:

$$\delta\mathbf{z} \triangleq \begin{bmatrix} \delta\boldsymbol{\omega}_1 \\ \delta\Omega_A \end{bmatrix} = \begin{bmatrix} {}^1\boldsymbol{\omega}_t^1 \\ \Omega_t^A \end{bmatrix} - \begin{bmatrix} {}^1\boldsymbol{\omega}^1 \\ \Omega_A \end{bmatrix} \triangleq \mathbf{z}_t - \mathbf{z} \tag{2.15}$$

The term $\delta\mathbf{z}$ is expressed as the difference between:

- $\mathbf{z}$ Measured angular velocities;
- $\mathbf{z}_t$ Reference angular velocities, that is, the target to be achieved.

In the reference velocity vector, the target angular velocity ${}^1\boldsymbol{\omega}_t^1$ is taken as the orbital angular velocity of the LVLH frame F_L relative to the inertial frame F_I.
At this point, the term representing the variation of the other measured variables is defined:

$$\delta(a) = \begin{bmatrix} \delta\mathbf{q}^{I,1} \\ \delta\varphi^A \end{bmatrix} \tag{2.16}$$

In this case, the quaternion error is represented as the Hamiltonian product between the measured quaternion and the target quaternion. The quaternion error is described in its multiplicative form:

$$\delta\mathbf{q}^{I,1} = \begin{bmatrix} \delta q_0^{I,1} \\ \delta\mathbf{q}_{vet}^{I,1} \end{bmatrix} = \mathbf{q}_t^{I,1*} \otimes \mathbf{q}^{I,1} \tag{2.17}$$

Vector component of quaternion is represent by $\mathbf{q}_{vet}^{I,1}$, while scalar component is represent by $q_0^{I,1}$. The second component of $\delta(a)$ is the azimuthal antenna error $\delta\varphi^A$, which represents

the difference between the measurement angle and the target angle of the antenna.
To develop the control law, it is necessary to take the time derivative of the sliding surface (derive Equation 2.14) and set it equal to a function that create the gradual transition to specifically condition, a function is:

$$\dot{\mathbf{s}} = \delta\dot{\mathbf{z}} + \mathbf{K}_2\delta\dot{\mathbf{a}} = -\mathbf{K}_1 \tanh(\eta\mathbf{s}) \tag{2.18}$$

where $\mathbf{K_1}$ and $\mathbf{K_2}$ are the gain matrix the first and the second order, while $\tanh(\eta\mathbf{s})$ is a transitional function. These matrix are diagonal and $\in \mathbb{R}^4$. The gain matrix is used to indicate the importance of error. The parameter η is the mitigation parameter of chattering. This phenomenon refers to rapid and repetitive oscillations around a desired parameter.
Using a first-order Taylor expansion, it's possible to rewrite the quaternion error and then find the derivative of its second part:

$$\delta\dot{\mathbf{a}} = \begin{bmatrix} \frac{1}{2}\left(\delta q_{I,1_0}\,\delta\boldsymbol{\omega}_1 + \delta\mathbf{q}_{I,1} \times (\delta\boldsymbol{\omega}_1 + 2\boldsymbol{\omega}_1)\right) \\ \delta\Omega_A \end{bmatrix} \tag{2.19}$$

The time derivative of the control state error, denoted as $\dot{\delta\mathbf{z}}$, requires knowledge of both the derivative of the target state, $\dot{\mathbf{z}}_t$, and the derivative of the current control state, $\dot{\mathbf{z}}$. The latter is obtained by explicitly deriving the spacecraft attitude dynamics, thus expressing $\dot{\mathbf{z}}$ as a function of the control state $\mathbf{z}$ and the control input $\mathbf{u}$.

To derive this expression, the linear and angular momentum equations are substituted into Equations 2.5, 2.6 and 2.7 representing the system dynamics, while adopting a simplified model of the spacecraft. In this model, both the antenna and the reaction wheels are assumed to be perfectly balanced, and no perturbations act on the spacecraft.

Since the necessary mathematical steps are quite extensive, the full derivation can be found in [11]. However, by developing these equations and rewriting them in matrix form, we arrive at the relation:

$$\mathbf{M}\dot{\mathbf{z}} = \mathbf{g} - \mathbf{u} \tag{2.20}$$

where the terms $\mathbf{M}$, $\mathbf{g}$, and $\mathbf{u}$ are also defined in Appendix B. Substituting this expression into Equation 2.19 and solving explicitly for $\mathbf{u}$ yields the control law:

$$\mathbf{u} = \mathbf{g} - \mathbf{M}\dot{\mathbf{z}}_t + \mathbf{K}_2\delta\dot{\mathbf{a}} + \mathbf{K}_1 \tanh(\eta\mathbf{s}) \tag{2.21}$$

The first three components of the control vector $\mathbf{u}$ represent the motor torques to be distributed among the five reaction wheels. The fourth component, denoted as $u_4 = -u(A)$, corresponds to the opposite of the motor torque applied to the antenna system. Since wheel 6 is designated to counteract the torques acting on the antenna, the torque applied to it will be:

$$u(6) = u_4 \tag{2.22}$$

While the motor torques acting on the other four reaction wheels will be:

$$\begin{bmatrix} u(2) & u(3) & u(4) & u(5) \end{bmatrix}^T = \mathbf{T}_{p,1}\,\mathbf{u}_{1:3} - u_4\,\hat{\mathbf{e}}_{16z} \tag{2.23}$$

where $\mathbf{T}_{p,1}$ is the $\mathbb{R}^{4\times3}$ matrix that converts from the F_1 coordinate system to the pyramidal configuration coordinates (see Figure 2.1).

2.2.4 Simulink model

All the previously described equations are implemented as *Simulink*'s subsystem blocks or MATLAB functions. Using *Simulink* software, dynamic systems can be model, simulated, and analysed visually through interconnected blocks, utilizing a graphical simulation environment integrated into MATLAB. The generic model is shown in Figure 2.3.
The AOCS model of the WIVERN spacecraft enables simulation of the system's dynamic and kinematic behavior.

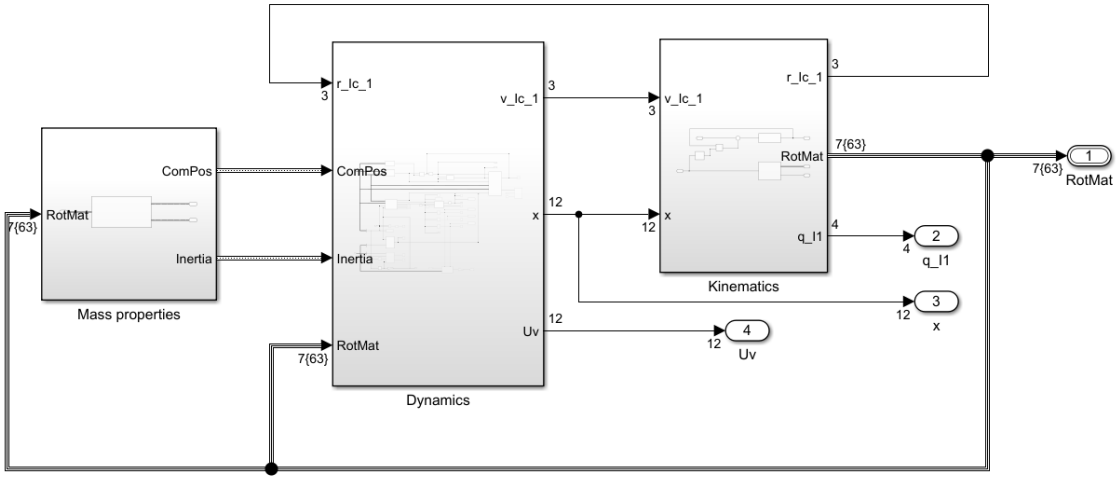

Figure 2.3: Observer out of the loop

An open-loop system uses input to produce output, but it doesn't use the output to adjust its behavior. This makes it easy to build, but less accurate, since it cannot correct errors or react to external changes. A closed-loop system uses feedback: compares the actual output with the desired value and adjusts the input to reduce the error.
All simulations are performed using a closed loop control system. However, to speed up the analysis and simplify the system, it is possible to choose not to use the filter's output (the estimate provided by the determination system) as input for the controller. This configuration will be referred to as observer-out-of-the-loop, where the controller uses the true state of the system, while the Kalman filter operates independently, without directly influencing the control.
The *Simulink* model shown in Figure 2.3 is divided into three main elements:

- **Mass Properties**: calculates the inertia characteristics of the system and the position of each component of WIVERN relative to F_1.

- **Dynamics**: calculate the velocities and control vector using the dynamic equations described above.

- **Kinematics**: calculates quaternion, the translational velocity vector of the main body, and the rotation matrix.

The update variables that are recalculated at each step are the position vector of the main body in the F_1 system and the rotation matrix.

2.2.5 Block 1: Mass Properties

The MassProp function is designed to calculate the mass and inertia properties of the spacecraft, the block represented in Figure 2.4. Its goal is to transform all the information related to the masses, positions of the center of mass, and inertia tensors of the individual components into the reference frame of the main body F_1. This process is necessary to obtain a consistent description of the overall dynamic behaviour of the vehicle.

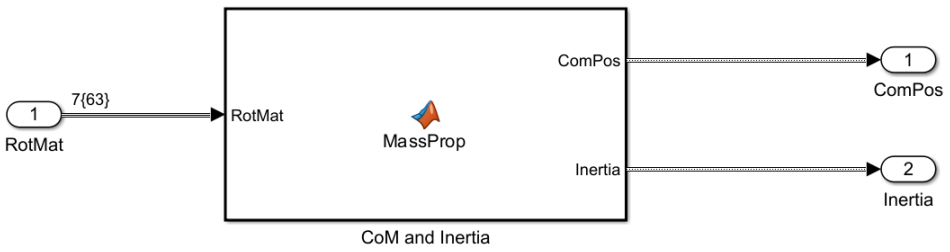

Figure 2.4: Mass Properties

The function accepts as input a set of relative position vectors, masses, inertia matrices, and rotation matrices from the reference frame of each component. The overall center of gravity of the vehicle is determined as a weighted average of the positions of the individual component centers of gravity. The rotation matrices are used to transform the local inertia tensors into the common reference frame. The function also allows for the calculation of mixed inertia terms, which account for the interactions between relative positions and mass distributions. The final output consists of two structures:

- **ComPos**: which collects the positions of the centers of mass;
- **Inertia**: which contains the transformed inertia tensors and the mixed terms.

This information is very important for the next steps of modelling, control, and simulation because it gives a full representation of how the system moves and behaves.

2.2.6 Block 2: Dynamics

The dynamics block is a complex system, each block of it will be examined separately. In this way, the function of each block and its importance within the system will be explained. The complete model is represented in Figure 2.5.

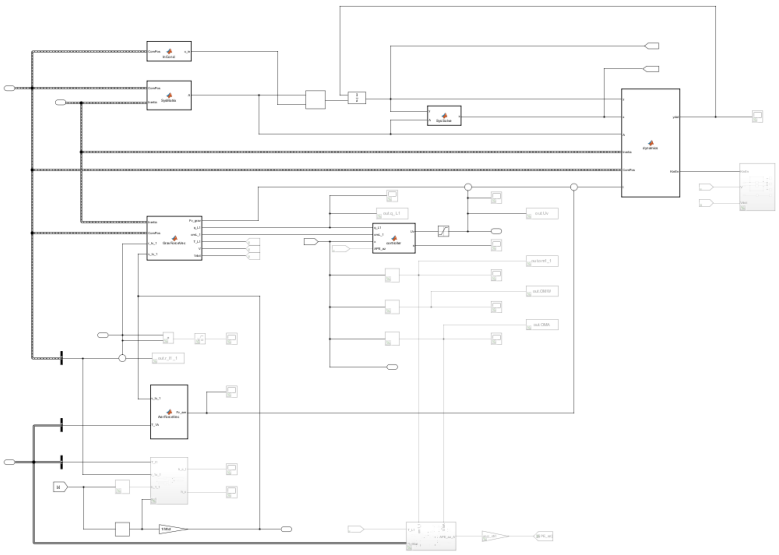

Figure 2.5: Dynamic block

Initial condition

The InCond function is made to calculate the starting speeds and angular velocities of the satellite and antenna. It takes input a set of parameters about the subsystems, including positions of the centers of mass and masses of each component and the initial angular speeds. First step, the function extracts the positions of the centers of mass of the various components. Then, it calculates the velocity of the center of mass of the main body, taking into account several contributions.
Each subsystem's velocity is determined by calculating the cross product between the component's center of mass and initial angular velocity:

$$\mathbf{v} = \mathbf{r} \times \omega \tag{2.24}$$

The final result is influenced by heavier subsystems because their contributions are weighted based on their relative mass.
Once all the angular velocity contributions are calculated, the function returns a vector x_{in} that contains the system's complete initial conditions.
This vector includes the velocity of the main body and the angular velocities of all subsystems.

$$\mathbf{v} = \frac{m}{M}(\mathbf{r} \times \omega) \tag{2.25}$$

Multibody system's matrix

The SysMatrix function creates the **A** matrix of Equation 2.4 through different steps. The first step is to extract the centers of mass and other components from the structure called ComPos. Next, it retrieves the inertia matrices and mixed inertia terms from the Inertia block. After collecting this information, the positions are transformed into matrices, which allow the calculation of vector products used in rigid body mechanics. Finally, the elements are assembled to construct the A matrix.

Block Gravity Force

The GravForceVec function is responsible for computing the gravitational forces and torques acting on the satellite.
As in the previous cases, the first step is to extract the necessary elements from the ComPos and Inertia structures. After that, the relative positions of each body with respect to the satellite's total center of mass are calculated. This step is important because gravitational forces do not act on a single point, but act on the entire body.
After preparing this data, the function builds the orbital reference frame LVLH (F_L). It also calculates the orbital velocity and orientation of the satellite with respect to F_L using quaternions. Finally, the function calculates gravitational forces and torques:

- A main force acts on the entire satellite, directed towards the center of the Earth:

$$f_1 = M_{\text{tot}} \cdot \frac{\mu_E}{r_{Ic}^2} \cdot \mathbf{e}_z^I \tag{2.26}$$

  Where M_{tot} is the sum of mass entire system, μ_E is the Earth's gravitational constant and $\mathbf{e}_z^I$ is the nadir-pointing unit vector.

- Each component experiences an additional force and torque, which depend on its position and mass.

All this information is combined into a generalized force vector, which includes both linear forces and torques.
Function calculates the gravitational potential energy of the system and its time derivative, which are useful for assessing the satellite's stability and the effectiveness of control strategies.

Block Aerodynamic Forces

The AerForceVec function is responsible for calculating the aerodynamic forces and torques acting on the satellite along its orbit.

Table 2.1: Orbital parameters

Parameter	Symbol	Value
Semi-major axis	a [km]	6878
Eccentricity	e	0
Inclination	i [deg]	97.418
Argument of perigee	ω [deg]	90
RAAN	Ω [deg]	120
Mean anomaly	M [deg]	0

The satellite is in a Low Earth Orbit (LEO) circular sun-synchronous orbit at an altitude of 500 km (orbital parameters in Table 2.1. In this type of orbit, atmospheric forces are still considerable. Each part of the satellite is affected differently, depending on shape, orientation, and position relative to the center of mass.
The function converts the positions of bodies 7 and 8 into the main reference frame F_1. It

then calculates the relative velocity of the satellite with respect to the atmosphere. This velocity is also expressed in the antenna's reference frame F_8, so that the drag effect on the curved surface can be evaluated correctly.
For the main body of the satellite, the function calculates the area that is actually exposed to the airflow. This depends on how the surfaces are oriented: only those facing the wind with a positive angle of incidence (positive cosine) contribute. A center of pressure is then estimated this is the average point where the force can be considered to act in order to create the same effect. Using this, the function calculates the aerodynamic force and the torque created by the distance between the center of pressure and the center of mass.
For the offset mass (m_7), a similar process is followed. The function applies a force based on the exposed area and velocity, at a known point relative to the center of mass of body 7.
The most complex case is the antenna (body 8). To compute the aerodynamic force on its curved surface, a 2D grid is used. For each grid element, the contribution to the airflow is calculated based on its orientation. The center of pressure of the reflector is also found in its own reference frame (F_8), then converted into the main reference frame F_1. As with bodies 1 and 7, both the aerodynamic force and the torque at a specific point are calculated.
Finally, all the aerodynamic torques are summed together to get the total torque in reference frame F_1. The output of the function is a generalized force vector that combines all the aerodynamic forces and torques acting on each satellite component.

Block dynamic equation

In this block, the various parts are represented in Figure 2.6, 2.7 and 2.8, all the elements connected to the outputs of the previously described blocks are used as inputs.
Matrix A is utilized in a multiplication block with the state vector from the Initial Conditions block at time t_0. A multiplication is performed between a column vector of size $[12 \times 1]$ and the matrix $A \in \mathbb{R}^{12\times12}$. At this point, the new vector x_0 passes through the Integrator block. This block performs the integral on the state variable to obtain the dependent variable y of the system, the vector that will be entered into the block represents the dynamic equations.

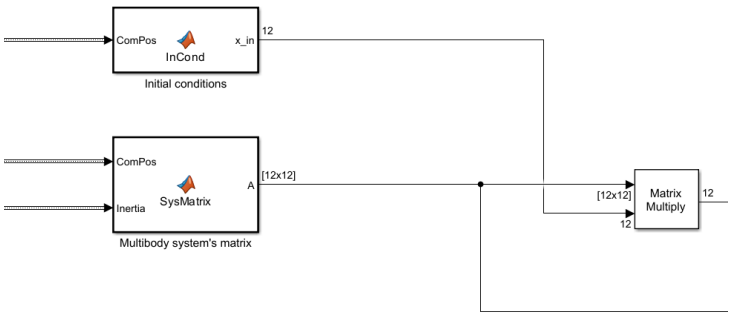

Figure 2.6: Initial condition and Multibody system's matrix block connection

The variable y, obtained through the Integrator block, enters a block that solves a

linear system:

$$x = A \backslash y \tag{2.27}$$

From this block, the solution to the differential equation is obtained.
The dynamics block also receives the variable t, which is given by the sum of gravitational and aerodynamic forces, plus the control vector ($\mathtt{U_v}$), limited through a Saturation block.

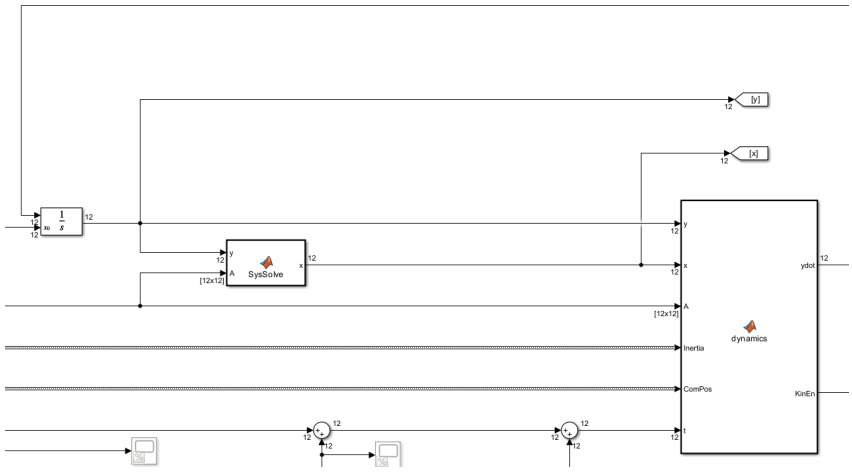

Figure 2.7: Dynamic Block

The Dynamics function represents the dynamic equations model. Its objective is to calculate the derivative of the system state and its kinetic energy, given the current state and the physical and geometric properties of the satellite. The system state x is decomposed into linear velocity, angular velocity of the main body, and the angular velocities of the various components. The momentum **p** and angular momentum **h** are also obtained.
Once these quantities are obtained, the derivatives of the dynamic state are calculated using Equations 2.5 and 2.6. The total forces and torques acting on the system (contained in the vector **t**) are used to determine the behavior of these quantities over time. The control vector is generated by the controller block, which will be described in the next section. This vector is updated based on x at each iteration.
The variable y, which is updated at each iteration, is used to obtain the velocity of the main body's center of mass in F_1.

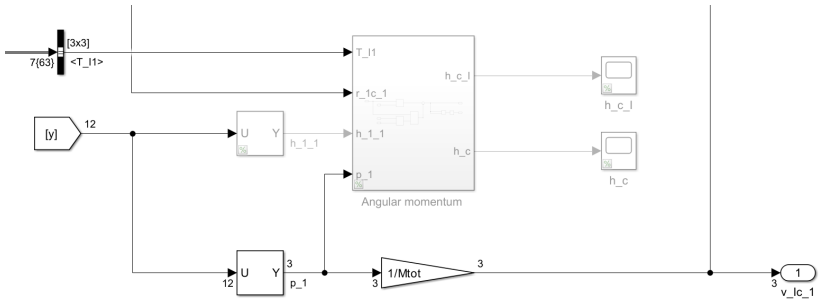

Figure 2.8: Output body one velocity

Finally, the function computes the total kinetic energy of the system. This allows monitoring of how much mechanical energy is associated with the satellite's internal dynamics, which is useful for stability analysis or evaluating control efficiency.

Control Block

The controller function is designed to implement the control commands needed to orient and stabilize the satellite. In this block represented in Figure 2.9, the sliding mode control described in the previous section is implemented. The code is primarily responsible for managing the satellite's attitude and reaction wheels, which are the main motors that control its rotation.
The function takes as input the attitude (through the quaternion), the angular velocities, and the state vector x. It then calculates the forces and moments acting on the satellite, accounting for both internal forces and external forces, such as those arising from the interaction between the satellite and the space environment. The control logic focuses on maintaining a stable orientation by correcting the errors between the satellite's current state and the desired one, using a sliding mode error based approach.
The code manages the desaturation of the reaction wheels. When the angular velocities of the wheels become too high, a desaturation strategy is activated to prevent damage to the wheels themselves.
The wheels are restored to safer angular velocity values using specific commands generated by the system when they detect that one of them has exceeded a certain velocity threshold.
The final result of the function is a vector containing the commands needed to correctly orient the satellite $\mathbf{U_v}$. This vector is then used to update the satellite's actuators.

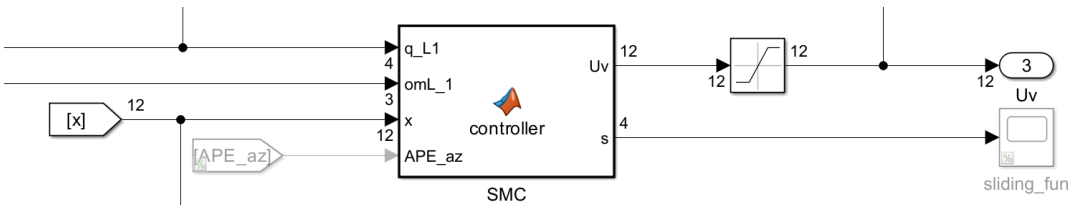

Figure 2.9: SMC block control

2.2.7 Block 3: Kinematics

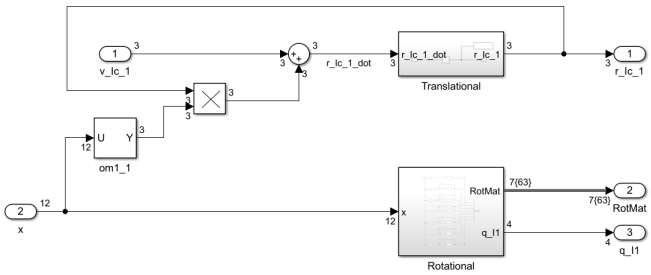

Figure 2.10: Kinematic Block

The kinematic block represented in Figure 2.10 receives as input the local velocity of the main body in F_1 and the state vector. This block implements the equations presented in the section related to the spacecraft's kinematics 2.2.2. The rotation matrix is the primary output of this block, which is used iteratively in the other blocks.
As a first step, the angular velocity components ω_1 of the main body of the satellite are extracted from the state vector. Subsequently, a cross product operation is performed between the position and ω_1, which allows for calculating the translational velocity of the satellite due to rotation. At this point, the two components of translational velocity are summed. This process is part of a closed loop, meaning it is executed at every iteration. Figure 2.10 illustrates the operation of this block. We will now proceed to describe the other blocks that comprise it.

Translational block

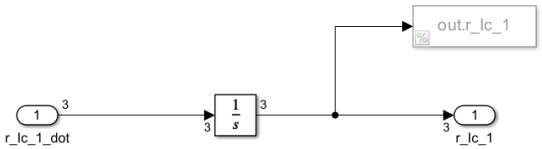

Figure 2.11: Traslation block

This block represented in Figure 2.11 is very simple: it takes as input the total translational velocity of the main body. This velocity represents the derivative of the main body's position. The output of this block is the updated position of the main body, calculated by considering the satellite's angular velocity.

Rotational block

This block takes as input the vector x, which contains the translational and angular velocities of each component. As output, the block provides the attitude quaternion in the reference frame F_1 and the rotation matrix for transforming between the various reference frames in the main frame F_1. To switch from one reference frame to another, the block described in Figure 2.13 is used.

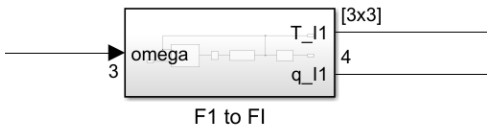

Figure 2.12: Simulink block "F1 to FI" used to compute the transformation between the body reference frame (F1) and the inertial frame (FI).

After extracting the relevant velocities from the vector x using a selector, two blocks can be encountered. The first, shown in Figure 2.12, takes ω_1 as input and provides the

rotation matrix for transforming from F_1 to F_I. The second, shown in Figure 2.14, is used to generate the rotation matrix from a system F_λ to F_1 (in the example, from F_2 to F_1). In the block shown in Figure 2.12, the quaternion derivative is implemented using Equation 2.10. For this reason, the QuatDerivative function takes the quaternion and ω_1 as inputs, and the output is the quaternion derivative.
The loop uses an initial value of the quaternion $\mathbf{q}_{t0}^{I,1}$, which allows the calculation through an integration block. The quaternion is the update variable that, after being normalized, re-enters the QuatDerivative block to perform the next iteration. To obtain the rotation matrix, the implementation of the Direction Cosine Matrix (DCM) is used:

$$T = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & q_1^2 - q_2^2 + q_3^2 - q_4^2 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & q_1^2 - q_2^2 - q_3^2 + q_4^2 \end{bmatrix} \tag{2.28}$$

Figure 2.13 shows the block that provides the quaternion and the rotation matrix from F_1 to F_I. The block shown in Figure 2.14 is similar to 2.12.

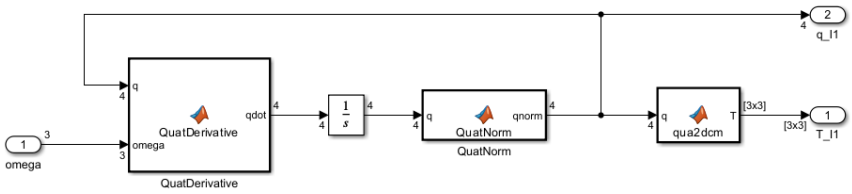

Figure 2.13: Simulink subsystem `"F2 to F1"`

The update variable is the rotation matrix itself. In this case, it is multiplied by the vector $\bar{\omega} = \mathbf{e}_\lambda{}^T\omega$ to obtain the angular velocity. This is used together with the quaternion to calculate the matrix at each step. The initial value of the quaternion depends on the two reference systems being considered.
This process of extracting the rotation matrix is performed for each reference frame, after which a single matrix is constructed, defined as RotMat.

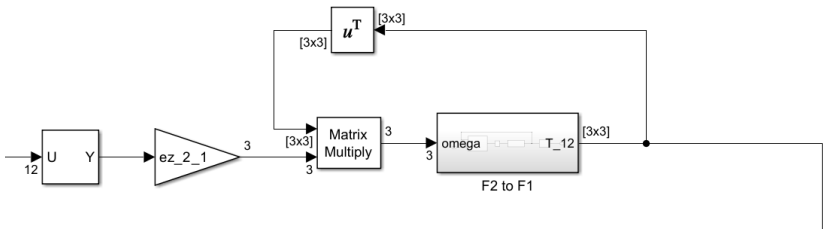

Figure 2.14: Details the internal structure of the transformation block.

Chapter 3

ADCS: Sensor model

This paragraph describes the theoretical model used to represent the sensors, measurements, and their associated noise. Additionally, as with the AOCS system, the implementation of the model in Simulink will also be analyzed.

3.1 Attitude Determination Architecture

The attitude determination system is essential for the WIVERN satellite, as high precision is required. It is important to ensure high accuracy in antenna pointing. The present attitude determination model is based on the work proposed in [11].

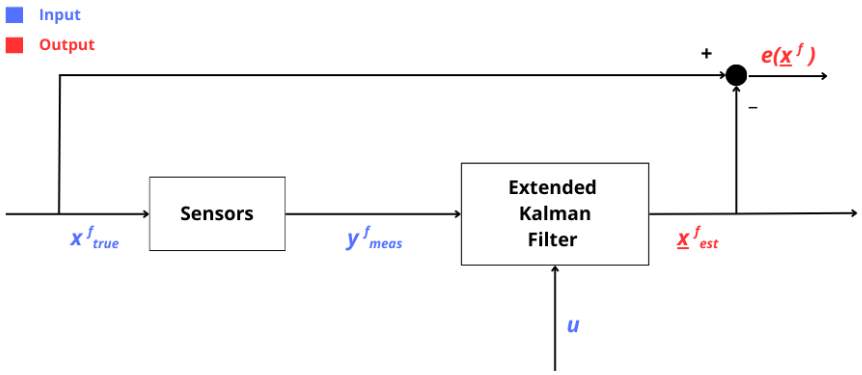

Figure 3.1: Schematic of the attitude determination architecture

In Figure 3.1, a schematic representation of the WIVERN attitude determination system is shown. The model consists of:

- **Sensors**: this block contains the complete model of the sensors used by WIVERN.
- **Extended Kalman Filter**: this block includes the filter that reduces the measurement error.

The sensors on the WIVERN satellite are responsible for providing an estimate of the main body's attitude, angular velocity, and the single-axis rotation of the antenna. The onboard sensor suite consists of three components:

- **Gyroscope**: measures the angular velocity of body 1 with respect to the F_I. A triaxial Fiber Optic Gyroscope (FOG) has been chosen.
- **Encoder**: measures the absolute angular position of the rotor of the antenna assembly with respect to body 1 at the hinge point O_A.
- **Star Tracker**: measures the spacecraft attitude in the inertial reference frame by observing a portion of the sky and identifying fixed stars. It has three star cameras (SC) that measure the rotation quaternion from the F_I frame to their respective local reference frames, defined based on their relative orientation to the main body. The measurements are combined to obtain a single attitude quaternion $\mathbf{q}^{\mathbf{I,1}}$.

A fundamental element in the determination model is the Kalman filter. The Extended Kalman Filter (EKF) is a standard method for nonlinear stochastic state estimation, commonly used in attitude determination and control systems. It is a recursive algorithm that uses a set of measurements and a system dynamics model to estimate the optimal state, with both sources of information impacted by random errors [11].
The EKF using a first-order Taylor series approximation to locally linearize the system equations at each time step, using Jacobian matrices to linearization complex non linear model. However, this linearization introduces limitations: the EKF's effectiveness is limited to systems with small non-linearity and does not take into account probabilistic uncertainties that arise from linearization itself. Despite these limitations, the EKF remains computationally efficient and easy to implement. Thanks to the knowledge of the current state and its covariance, it allows predicting the next state and its associated uncertainty, ensuring a robust state estimation [14].
In the next chapter, the EKF and its implementation in Simulink will be analyzed.

3.2 Sensors models

In this section, the mathematical and physical models describing the sensors are presented, which are used for implementation within the simulator.
Each real measurement is considered as the sum of two components as described by Equation 3.1.

$$\mathbf{y}^f_{meas} = \mathbf{x}^f_{true} + \boldsymbol{\nu}_{rumor} \tag{3.1}$$

The description of the two components is provided below:

1. **True component** $\mathbf{x}^f_{true}$: the part of the measurement not affected by instrument errors.
2. **Disturbance component** $\boldsymbol{\nu}_{rumor}$: the part related to measurement disturbances that generate deviations from the correct measurement.

Each measurement instrument has a value of noise that is influenced by various factors, which can be predicted using specific theoretical models.

3.2.1 Gyroscope

The three-axis gyroscope model that describes the noise generated by this instrument is as follows:

$$^1\boldsymbol{\omega}^1_{\text{meas}} = {}^1\boldsymbol{\omega}^1 + \mathbf{G}_\omega \mathbf{T}_\omega {}^1\boldsymbol{\omega}^1 + \mathbf{w}_\omega + \mathbf{b}_\omega + \mathbf{n}_\omega \tag{3.2}$$

The components that form the equation are:

- ${}^1\boldsymbol{\omega}^1$: is the angular velocity of body 1 in the reference frame F_1, obtained from the equation of motion.

- $\mathbf{G}_\omega$: is a scale matrix representing the scale error. The scale error is mainly caused by the instrument not properly converting the physical signal into a digital (electrical) signal. This causes an incorrect proportionality between the data and the measurement actually collected. This component is given by $\mathbf{G}_\omega = \Delta\theta_\omega \mathbf{E}3$ where $\mathbf{E}3$ is an identity matrix $[3 \times 3]$ and $\Delta\theta_\omega$ is a misalignment angle.

- $\mathbf{T}_\omega$: represents a rotation matrix around the three axes. The rotation angle of the axes is also the misalignment angle, $\Delta\theta_\omega$ [11].

- $\mathbf{w}_\omega$: represents white noise, interpretable as random error, and is described by a Gaussian distribution. The Gaussian distribution is a continuous probability distribution that describes how the values of a random variable are distributed around a mean. Its shape is that of a symmetric "bell curve" [16]. An important element for describing a Gaussian distribution is **Covariance** (σ^2), which is the **Standard Deviation** (σ) squared.

$$\sigma_\omega^2 = \frac{W_\omega^2}{f_\omega} \tag{3.3}$$

  Where W_ω^2 is the Angular Random Walk (ARW) of the gyroscope, while f_ω is the sensor sampling frequency.

- $\mathbf{b}_\omega$: represents the systematic errors of the gyroscope, which is modeled with two components, one static and one time changing:

$$\mathbf{b}_\omega = b_0 + \int_{t_0}^{t_1} \eta_\beta \, dt \tag{3.4}$$

  Where:

  - b_0 is the constant initial bias;
  - The second component is the variation component respect to the constant initial value over a time interval Δt. It is modeled as a first-order random walk with η_β as Gaussian white noise with covariance:

$$\sigma_\eta^2 = \beta_\omega^2 \frac{f_\omega}{\Delta t_\beta} \tag{3.5}$$

- $\mathbf{n}_\omega$: this term represents the error due to signal discretization and depends on the number of bits $\mathbf{N}_{bits}^\omega$.

The specifications of the gyroscope are reported in Table 3.1.

Parameter	Symbol	Value
Sample rate	f_ω	200 Hz
Full scale	$\pm\omega_{\max}$	$\pm$15 [deg/s]
Misalignment	$\Delta\theta_\omega$	25 μrad
Scale error max	$\Delta\omega$	40 ppm
ARW	W_ω	10^{-4} [deg/$\sqrt{h}$]
Bias stability over 1 h	β_ω	2×10^{-4} [deg/h]
Resolution	N_ω	32 bits

Table 3.1: Sensor Characteristics

3.2.2 Rotary Encoder

The rotary encoder measures the rotation angle around the axis $\hat{\mathbf{e}}_{A_z}$. The measured absolute angular position $\varphi_{A_{\text{meas}}}$, is expressed as:

$$\boldsymbol{\varphi}^A_{\text{meas}} = \boldsymbol{\varphi}^A_{\text{true}} + \mathbf{w}_\varphi + \mathbf{n}_\varphi \tag{3.6}$$

The three terms represent, respectively:

- $\boldsymbol{\varphi}^A_{\text{true}}$: The true absolute angular position of rotation;
- $\mathbf{w}_\varphi$: represents the random error, modeled with a Gaussian distribution and covariance σ^2_{enc};
- $\mathbf{n}_\varphi$: represents the error due to signal discretization, as in the case of the gyroscope.

Table 3.2 presents the specifications of the encoder.

Parameter	Symbol	Value
Sample rate	f_{enc}	8 [kHz]
Radius	r_{enc}	10.16 [cm]
Resolution	$N_{\text{enc, bits}}$	25 [bits]
White noise covariance	σ_{enc}	0.64 [arcsec^2]

Table 3.2: Encoder Characteristics

3.2.3 Star tracker

The onboard Star tracker consists of three star cameras, which will subsequently be identified with $\alpha = 1$, 2, and 3. The attitude quaternion $\mathbf{q}^{I,1}_{\alpha,\text{meas}}$ can be expressed using the Hamiltonian product, which is the product of quaternions:

$$\mathbf{q}^{I,1}\alpha, \text{meas} = \mathbf{q}^{I,1}\alpha, \text{true} \otimes \mathbf{q}^{I,1}_{\alpha,\text{err}} \tag{3.7}$$

The measurement error quaternion describes a small angle to the point where it can be expressed as:

$$\mathbf{q}^{I,1}_{\alpha,\text{err}} \approx \left[1, \frac{1}{2}\,{}^{\alpha}\varepsilon_\alpha\right] \tag{3.8}$$

The vector of small angles $\boldsymbol{\varepsilon_\alpha}$ is expressed in the camera's reference frame α. To describe it in the main body frame, it must be multiplied by the rotation matrix $\mathbf{T}^{1,\alpha}$:

$$^{1}\boldsymbol{\varepsilon_\alpha} = \mathbf{T}^{1,\alpha}\boldsymbol{\varepsilon_\alpha} \tag{3.9}$$

The component $\boldsymbol{\varepsilon}$ is composed of the following terms:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_{bias} + \boldsymbol{\varepsilon}_{TE}(T) + \boldsymbol{\varepsilon}_{FSE} + \boldsymbol{\varepsilon}_{PSE} + \boldsymbol{\varepsilon}_{TN} \tag{3.10}$$

Where:

- $\boldsymbol{\varepsilon}_{\text{bias}}$ is the bias term related to systematic errors. This can be due to calibration residuals, launch-induced misalignment, and degradation.

- $\boldsymbol{\varepsilon}_{TE}(T) = \boldsymbol{\varepsilon}_{TE,T}\Delta T$ represents temperature variation errors. This term depends on $\Delta T = T - T_0$, where T_0 is the average value of the range. Since there is no thermal model, the worst-case scenario ΔT_{max} is used, thus $\boldsymbol{\varepsilon}_{TE}^{max}$. The maximum systematic bias error is:

$$\boldsymbol{\varepsilon}_{sys}^{max} = \boldsymbol{\varepsilon}_{bias} + \boldsymbol{\varepsilon}_{TE}^{max} \tag{3.11}$$

- The terms $\boldsymbol{\varepsilon}_{FSE}$ and $\boldsymbol{\varepsilon}_{PSE}$ represent the Field of View error (FOV), spatial error (FSE) and pixel error (PSE). These are modeled as first-order Gauss-Markov processes:

$$\dot{\boldsymbol{\varepsilon}}_s = \tau_s^{-1}\varepsilon_s + w_s \tag{3.12}$$

- $\boldsymbol{\varepsilon}_{TN}$ represents temporal noise, modeled as zero-mean Gaussian white noise with covariance $\sigma^2 TN$.

Star tracker errors are represented by three angles around the camera's reference frame axes, where the Z-axis corresponds to the line of sight (boresight axis). The main errors are σ_{FSE} (Field of View Spatial Error), σ_{PSE} (Pixel Spatial Error), and σ_{TN} (Temporal Noise).

The system uses three distinct star cameras. The optimal combination of quaternions measured by each camera is obtained using a least squares approximation with a weighting matrix [17] [11].

The three star cameras are oriented to avoid direct view of the Earth and the Sun. Specifically, Camera 1 is pointed towards the zenith, while the other two are oriented with a rotation of $\pm 90°$ around the y axis and $15°$ around the x axis.

The analysis considers the operational specifications of the star cameras and their arrangement, highlighting how the optimized orientation improves the overall system accuracy [11].

The characteristics of the star tracker are summarized in Table 3.3.

Parameter	Value
Sampling Frequency (f_{ST})	20 [Hz]
Star Cameras (SCs) α	1, 2, 3
Field of View (FOV)	23 [deg]
Pixels (N_{pixel} across)	1024
Operational Temperature ($T_{\min}, T_{\max}$)	-30, +60 [°C]
Bias Error (ϵ_{bias}) (all axes)	11 [arcsec]
Thermal Error ($\epsilon_{TE,T}$) (all axes)	0.055 [arcsec/K]
σ_{FSE} across Z	0.2 [arcsec]
σ_{FSE} around Z	1.53 [arcsec]
σ_{PSE} across Z	1.33 [arcsec]
σ_{PSE} around Z	9 [arcsec]
σ_{TN} across Z	0.77 [arcsec]
σ_{TN} around Z	6 [arcsec]
Earth Exclusion Angle (EEA)	18.5 [deg]

Table 3.3: Star Tracker Characteristics

3.3 Simulink model

The complete Simulink model of the sensors includes the implementation of a model for each sensor, as shown in Figure 3.2. This section describes only the model related to the sensors.

3.3.1 Gyroscope Simulink model

Like in Equation 3.2, four disturbance components affect the measurement. The gyroscope block takes three inputs:

- *om_true*, which is $\boldsymbol{\omega}_1^1$.
- *om_noise* is modeled as white noise, generated using a Band-Limited White Noise block. The noise power is defined as:

$$\mathbf{W}_\omega = \begin{bmatrix} W_\omega^2 \\ W_\omega^2 \\ W_\omega^2 \end{bmatrix} \tag{3.13}$$

  which is the vector representing the Angular Random Walk.

- *b_gyro* represents the systematic error, modeled by a random number generator with zero mean and a covariance of:

$$\frac{\beta_\omega^2 \cdot f_\omega}{\beta_T} \tag{3.14}$$

  where β_T is the time over which β_ω is defined. The generated signal is then integrated over time using an integration block, accurately simulating the cumulative effect of the bias.

The *gyro* function simulates the operation of a gyroscope by adding noise components directly to the true angular velocity. Two additional noise components are:

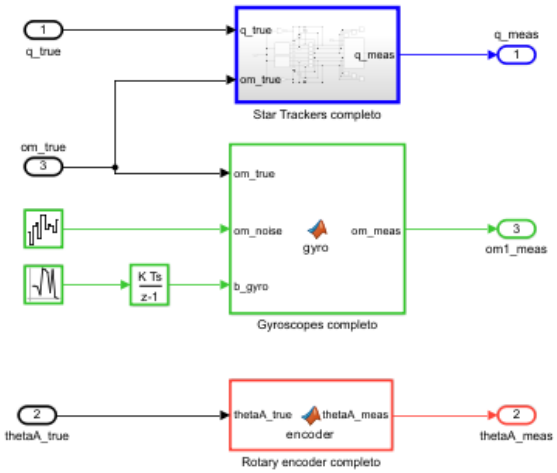

Figure 3.2: Sensor Simulink model

- Scale error, defined as a scaling of the true angular velocity using a scale error matrix and rotation matrix.
- Signal discretization error, modeled as quantization:

$$\omega_{meas} = \Delta\omega \cdot \left(\frac{\omega_{\mathbf{sens}}}{\Delta\omega} \right) \tag{3.15}$$

The final output is the measured variable, ω_{meas}.

3.3.2 Rotary Encoder Simulink model

The encoder model has a single input, the true rotation angle φ_A^{true}. It simulates an angular encoder with noise and geometric correction:

- Calculation of the geometric correction angle (β).
- Addition of zero-mean Gaussian white noise.
- Calculation of the measured angle:

$$\theta_{sens} = \varphi_A^{true} + \beta - \beta_0 + w_\varphi \tag{3.16}$$

The final output is the measured angle, φ_{meas}^A.

3.3.3 Startracker Simulink model

The Simulink model implementing the Startracker is based on the quaternion representation model, as shown in Equation 3.7. This model rapresented in Figure 3.3 adopts the multiplicative quaternion representation. Quaternion errors cannot be treated like the

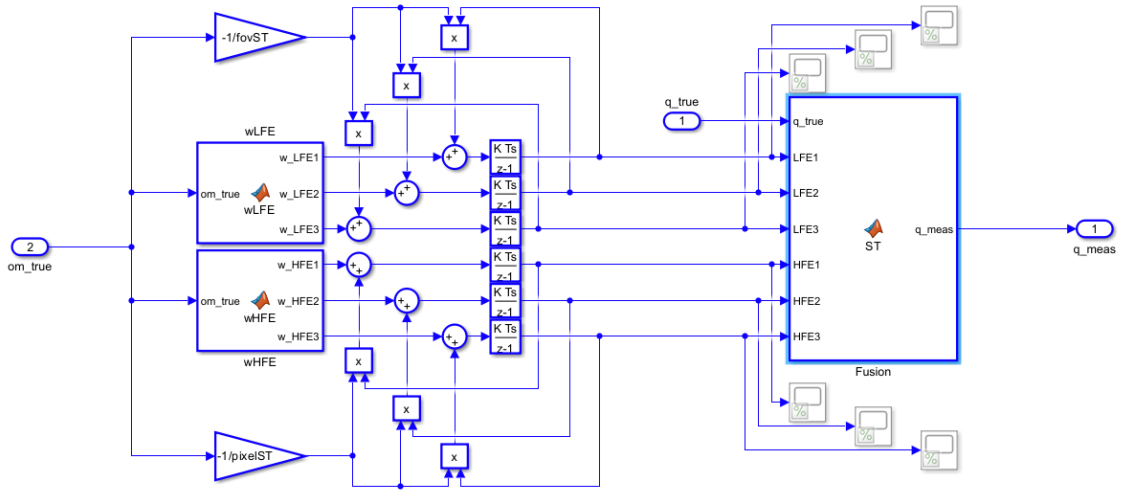

Figure 3.3: Simulink model startracker

other two sensors because quaternions are a different physical entity and cannot be handled as simple vectors.

The inputs of the model are the quaternion (*q_true*) and the angular velocity (*om_true*) of the body 1 calculated by the AOCS system. Starting from these inputs, the measured quaternion is calculated, which is the only output of the system described in Figure 3.3.

- The input *om_true* is divided into two main paths: one for the Low-Frequency Error (LFE) and one for the High-Frequency Error (HFE).
- LFE and HFE functions generate angular errors for the three star trackers, considering the rotation matrix T_{1ST_i}. LFE is linked to the Field of View (FOV) precision, while HFE depends on quantization errors linked to the number of pixels.
- These errors are combined to generate three components for each path: LFE1, LFE2, LFE3 for LFE and HFE1, HFE2, HFE3 for HFE.
- Each component is further processed using gain and delay blocks to simulate system dynamics.
- The signals are then combined with the reference quaternion (q_{true}) to generate the measured quaternion (q_{meas}).

Finally, the measured quaternion is obtained by normalizing the quaternion to maintain its unit norm, ensuring it remains a valid quaternion representation.

Chapter 4

Filter design

4.1 Filtering model

In this section, the state of the art of the filtering model will be described. The first implemented solution, which has already produced excellent results, is considered the baseline model for developing an improved solution. Specifically, the initial solution, as previously mentioned, is an Extended Kalman Filter (EKF). Furthermore, this paragraph will describe the mathematical model of the Unscented Kalman Filter (UKF) and the main differences between the two filters.

4.1.1 Kalman Filter

In 1960, R.E. Kalman invented an innovative method to recursively solve the problem of linear filtering of discrete data. The Kalman filter has become a central topic of research and development due to its applicability in various fields.

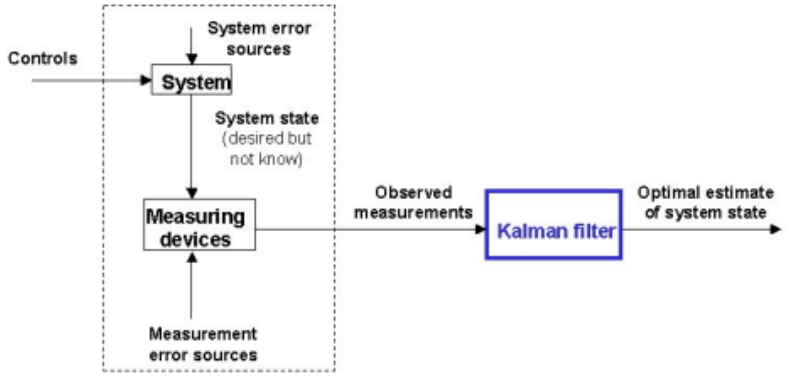

Figure 4.1: Typical application of the Kalman Filter [credit [18]]

Figure 4.1, reproduced from [18], illustrates the application context in which the Kalman Filter is used. A physical system, (e.g., a mobile robot, a chemical process, a satellite) is driven by a set of external inputs or controls and its outputs are evaluated by measuring devices or sensors, such that the knowledge on the system's behavior is solely given by the inputs and the observed outputs. The observations convey the errors and

uncertainties in the process, namely the sensor noise and the system errors. Based on the available information (control inputs and observations) it is required to obtain an estimate of the system's state that optimizes a given criteria. This is the role played by a filter [18]. The Kalman filter is an algorithm based on a set of mathematical equations that provides a recursive and efficient solution to the least squares method. Its effectiveness lies in its ability to estimate not only the current state of a system but also past and future states, even in the presence of uncertainties in the system model itself. [19]
Kalman filters are a family of filtering algorithms used to estimate the state of a dynamic system in real-time. When the system state dynamics or observation dynamics are nonlinear, the associated probability distributions are non-Gaussian, making optimal filtering computationally complex [20].

4.1.2 Extended Kalman Filter (EKF)

The Extended Kalman Filter (EKF) addresses this problem by linearizing the nonlinear dynamics around the previous state estimates, allowing the standard Kalman filter model to be applied. This approximation provides a state estimate with reduced computational cost compared to a full nonlinear filter. [20]
To introduce the EKF algorithm, a general representation of a dynamic system in discrete-time at instant t_k is used, as described in equations 4.1 and 4.2:

$$\mathbf{x}_k = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{d}_{x_k} \tag{4.1}$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{d}_{y_k} \tag{4.2}$$

The main goal of a filtering algorithm is to accurately estimate the true system state $\mathbf{x}_k$ at time t_k, even though it is not directly observable. The estimate $\hat{\mathbf{x}}_k$ is computed using the previous estimates $\hat{\mathbf{x}}_{k-1}$ and the known values of the input $\mathbf{u}_k$ and output $\mathbf{y}_k$ vectors. The terms $\mathbf{d}_{x_k}$ and $\mathbf{d}_{y_k}$ represent the disturbances associated with the state and output. These disturbances are modeled as zero-mean Gaussian white noise. The functions f and h are the state and measurement functions, respectively. These functions map the state vector and the inputs to the system outputs. After choosing an initial state estimate $\hat{\mathbf{x}}_0$ and initial estimation error covariance matrix $\mathbf{P}_0$ [11].
The filter EKF's algorithm is separate in two main phases:

1. **Prediction**: The prediction of the state estimate and its covariance matrix is performed using data from the previous step. For integration, a forward Euler method is used, with τ_f denoting the sampling interval of the filter. The matrix $\mathbf{Q}$ denotes the covariance of the process noise [21]:

$$\mathbf{x}_k^p = \hat{\mathbf{x}}_{k-1} + \tau_f f\left(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}\right) \tag{4.3}$$

$$\mathbf{P}_k^p = \mathbf{F}_{k-1}\mathbf{P}_{k-1}\mathbf{F}_{k-1}^\top \tag{4.4}$$

2. **Update**: The state estimate and covariance matrix are updated based on the prediction and on current information from the measurements. In this step, the Kalman gain $\mathbf{K}_k$ is computed, chosen to minimize the estimation error norm $\|\mathbf{x}_k - \hat{\mathbf{x}}_k\|^2$ [11]:

$$\mathbf{K}_k = \mathbf{P}_k^p \mathbf{H}_k^\top \left(\mathbf{H}_k \mathbf{P}_k^p \mathbf{H}_k^\top + \mathbf{R}_k \right)^{-1} \tag{4.5}$$

$$\Delta \mathbf{y}_k = \mathbf{y}_k - h(\mathbf{x}_k^p) \tag{4.6}$$

$$\hat{\mathbf{x}}_k = \mathbf{x}_k^p + \mathbf{K}_k \Delta \mathbf{y}_k \tag{4.7}$$

$$\mathbf{P}_k = (\mathbf{E}_n - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^p \tag{4.8}$$

$\mathbf{R}$ is the measurement noise covariance matrix. For this study, the state and measurement vectors, respectively $\mathbf{x}_f$ and $\mathbf{y}_f$ are defined as:

$$\mathbf{x}_f = \begin{bmatrix} \mathbf{q}_{I,1} & \phi_A & \boldsymbol{\omega}_1 & \Omega_A \end{bmatrix}^\top \tag{4.9}$$

$$\mathbf{y}_f = \begin{bmatrix} \mathbf{q}_{I,1}^{\text{meas}} & \phi_A^{\text{meas}} & \boldsymbol{\omega}_1^{\text{meas}} \end{bmatrix}^\top \tag{4.10}$$

The function h is simply defined as:

$$h(\mathbf{x}_f) = \begin{bmatrix} \mathbf{q}_{I,1} & \phi_A & \boldsymbol{\omega}_1 \end{bmatrix}^\top \tag{4.11}$$

Consequently, the measurement matrix $\mathbf{H}$, derived as the Jacobian of the function h evaluated at the estimated state $\hat{\mathbf{x}}_k$, results in a constant matrix of dimensions $[8 \times 9]$.
It is important to observe that Ω_A is included in the estimation process because it is required during the state propagation phase. However, since it is not a directly measurable quantity, its value is inferred internally by the EKF algorithm. At this point, it is necessary to define the function f, from which the state transition matrix $\mathbf{F}_k$ is obtained.

$$\mathbf{F}_k \triangleq \mathbf{I} + \tau_f \left. \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k} \tag{4.12}$$

where $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ is the Jacobian of f evaluated at $\hat{\mathbf{x}}_k$.
This approach is similar to the one found in for the filter algorithm. The complete form of the function f accounts for both the rotation of the antenna assembly and the quaternion kinematics in the time evolution of the filter states [11].

$$\dot{\mathbf{x}}^f = \begin{cases} \dot{\mathbf{q}}^{I,1} \\ \dot{\phi}^A \\ \dot{\boldsymbol{\omega}}^1 \\ \dot{\boldsymbol{\Omega}}^A \end{cases} = \begin{cases} \frac{1}{2}\mathbf{B}(\mathbf{q}^{I,1})\,\boldsymbol{\omega}^1 \\ \Omega^A \\ \mathbf{M}^{-1}(\mathbf{g} - \mathbf{u}) \end{cases} = f(\mathbf{x}^f, \mathbf{u}, \boldsymbol{\Gamma}) \tag{4.13}$$

where $\boldsymbol{\Gamma} = \{\Omega^2, \ldots, \Omega^5\}^T$ is a vector containing the known reaction wheels angular velocities [11].

The EKF algorithm can be summarized using the block diagram shown in Figure 4.2 [12]:

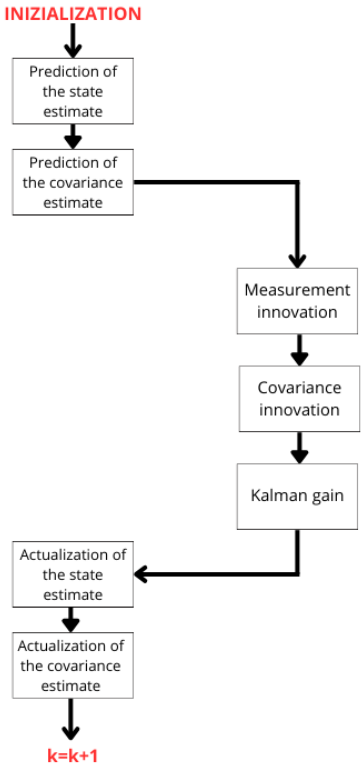

Figure 4.2: Block scheme to explain the algorithm

Table 4.1: EKF Parameters

Parameter	Value
f_{ekf}	10 [Hz]
$\mathbf{Q}$	$10^{-11} \times \text{diag}(1,\ 1,\ 1,\ 1,\ 1,\ 100,\ 100,\ 100,\ 1)$
$\mathbf{R}$	$10^{-9} \times \text{diag}(10,\ 10,\ 10,\ 10,\ 0.1,\ 0.1,\ 0.1,\ 1)$
$\mathbf{H}$	$[\mathbf{E}_8 \quad \mathbf{0}_{8\times 1}]$
$\mathbf{x}_f^0$	$\begin{bmatrix} 1 & \mathbf{0}_{7\times 1} & \Omega_6^0 \end{bmatrix}^T$
$\mathbf{P}_0$	$\mathbf{E}_9$

The parameters of the filter are presented in Table 4.1.
Identity matrices are denoted by $\mathbf{E}_n$, which represents an identity matrix of dimension $n \times n$, while the zero matrix or zero vector is represented by $\mathbf{0}_{m\times n}$, indicating a matrix of size m by n.
These values are considered as initialization parameters, meaning they are used for the initial trials. Subsequently, they will be adjusted to achieve the values that yield the best results.

4.2 Simulink model

The complete model of the determination system includes the EKF filter and the sensor model, as shown in Figure 4.3. Several blocks are present, whose role is to generate outputs in the form of graphical representations. These components do not contribute directly to the numerical output generation; instead, they serve to coherently represent the outputs graphically.

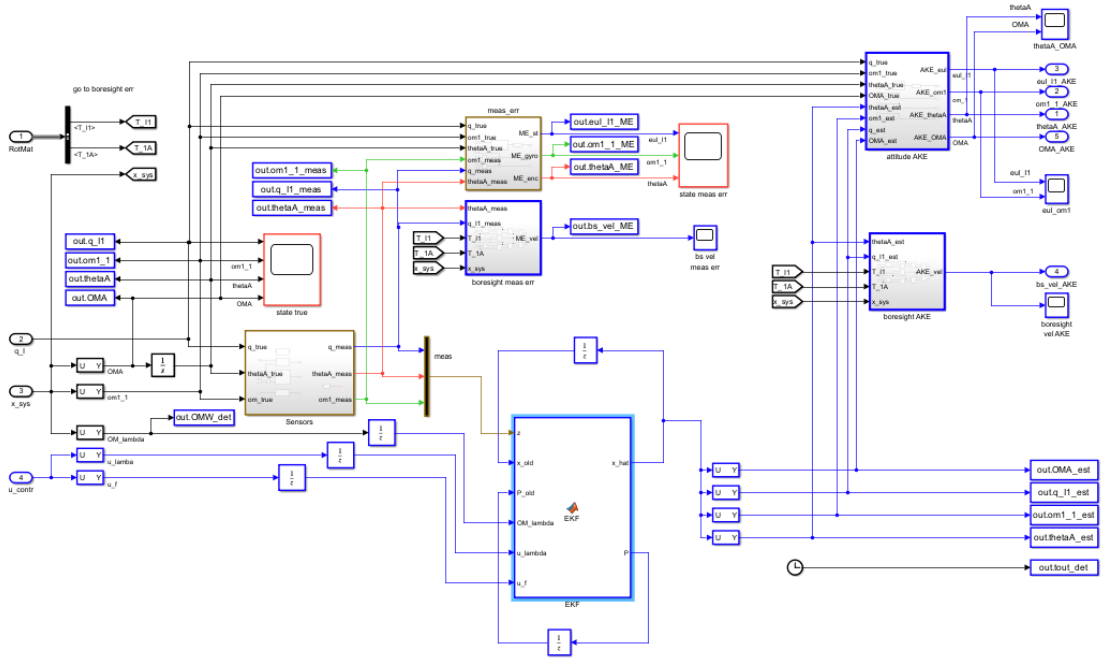

Figure 4.3: Determination system model

4.2.1 Filter model

The algorithm takes as input the previously estimated state $\mathbf{x}_{\text{old}}$, the previous state error covariance matrix $\mathbf{P}_{\text{old}}$, current measurements $\mathbf{z}$, and a set of physical and geometric parameters describing the structure and the acting forces on the system. The output of the filter consists of the updated state estimate $\hat{\mathbf{x}}$ and the corresponding error covariance matrix $\mathbf{P}$.

At the core of the algorithm lies a nonlinear function $f(\mathbf{x})$ that models the system dynamics. This function is composed of two main parts:

- the **kinematics** of the attitude quaternion $\mathbf{q}_{I1}$ and the antenna angle φ_A;
- the **rotational dynamics**, which take into account the forces and torques generated by the various subsystems (actuators, distributed masses, inertias) and the interactions between the main body and the antenna.

To handle the nonlinearity of the system, the algorithm linearizes the function $f(\mathbf{x})$ around the current estimated state. This linearization allows the computation of the state transition matrix $\mathbf{F}_k$, which is required to update the prediction of the covariance.

The EKF then proceeds through its two standard phases:

1. **Prediction**
   In this phase, the state is propagated forward in time using an explicit Euler integration scheme. Simultaneously, the error covariance matrix is updated. This step represents the expected system behavior based on the physical model and previous information.

2. **Update (Correction)**
   Once new measurements are received, they are compared to the predicted values. The difference (called the innovation) is used to correct the state estimate using the Kalman gain. The correction accounts for both the model uncertainty (expressed by $\mathbf{Q}$) and the measurement noise (expressed by $\mathbf{R}$).

This entire process is repeated at each sampling instant, enabling continuous and adaptive state estimation even in the presence of noise and disturbances. The code used to implement the filter inside a Simulink block is as follows:

Algorithm 1 Extended Kalman Filter (EKF)

function EKF($\mathbf{z}, \mathbf{x}_{\text{old}}, \mathbf{P}_{\text{old}}, \Omega_\lambda, \mathbf{u}_\lambda, \mathbf{u}_f, \tau_f, \mathbf{H}, \mathbf{Q}, \mathbf{R}, M_{\text{tot}}, M_A, \mathbf{r}_{1A,1}, S_{\text{mod}}, \mathbf{e}_{z_2}, \ldots, \mathbf{e}_{z_6}, \mathbf{e}_{z_A}$)
  Define system dynamics function $f(\mathbf{x})$:
  - Quaternion kinematics:
    $$\dot{\mathbf{q}}_{I1} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_1)\mathbf{q}_{I1}$$
  - Antenna angle dynamics:
    $$\dot{\varphi}_A = \Omega_A$$
  - Angular velocity dynamics using rigid-body model and actuation:
    $$\dot{\boldsymbol{\omega}}_1,\ \dot{\Omega}_A = \text{function of } \mathbf{u}_\lambda, \Omega_\lambda, S_{\text{mod}}, \mathbf{e}_{z_i}$$

  Compute Jacobian $\mathbf{F}_k$:
  $$\mathbf{F}_k = \mathbf{I}_9 + \tau_f \cdot \left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_{\text{old}}}$$
  Prediction step:
  $\mathbf{x}_{\text{pred}} = \mathbf{x}_{\text{old}} + \tau_f \cdot \left(f(\mathbf{x}_{\text{old}}) + \begin{bmatrix} \mathbf{0}_{5\times 1} \\ S_{\text{mod}}^{-1}\mathbf{u}_f \end{bmatrix} \right)$
  $\mathbf{P}_{\text{pred}} = \mathbf{F}_k \mathbf{P}_{\text{old}} \mathbf{F}_k^\top + \mathbf{Q}$
  Update step:
  $\mathbf{S} = \mathbf{H}\mathbf{P}_{\text{pred}}\mathbf{H}^\top + \mathbf{R}$
  $\mathbf{K} = \mathbf{P}_{\text{pred}}\mathbf{H}^\top\mathbf{S}^{-1}$
  $\Delta\mathbf{z} = \mathbf{z} - \mathbf{H}\mathbf{x}_{\text{pred}}$
  $\hat{\mathbf{x}} = \mathbf{x}_{\text{pred}} + \mathbf{K}\Delta\mathbf{z}$
  $\mathbf{P} = (\mathbf{E}_9 - \mathbf{K}\mathbf{H})\mathbf{P}_{\text{pred}}$
  return $\hat{\mathbf{x}}, \mathbf{P}$
end function

The model used for the final simulations includes the integration of the filter's output variables within the AOCS model. This configuration shown in Figure 4.4, known as the

observer in the loop, uses the estimated state for control purposes. In this case, the Kalman filter operates in cooperation with the control system, and the estimation error directly affects the control response. This setup more accurately reflects the behavior of the real system, as it simulates the actual interaction between estimation and control. Conversely, the observer out of the loop configuration was employed during the development phase to analyze the performance of the attitude determination model in isolation, providing a clear view of its behavior without interference from control dynamics.
The simulation model can be described through the following components:

- **T_11est, om1_1_est, OMA_est**
  These quantities are computed within the Determination block and represent the estimated torque and angular velocities of the system. They are directly used in the Dynamics block for computing the equations of motion.

- **x_sys**
  This is the estimated state of the system, used as a predictive input in subsequent simulation steps, allowing the model to dynamically adapt to current conditions.

- **RotMat**
  The rotation matrix is shared among multiple blocks and is essential to maintain the orientation of the system across all kinematic and dynamic transformations, ensuring spatial consistency within the model.

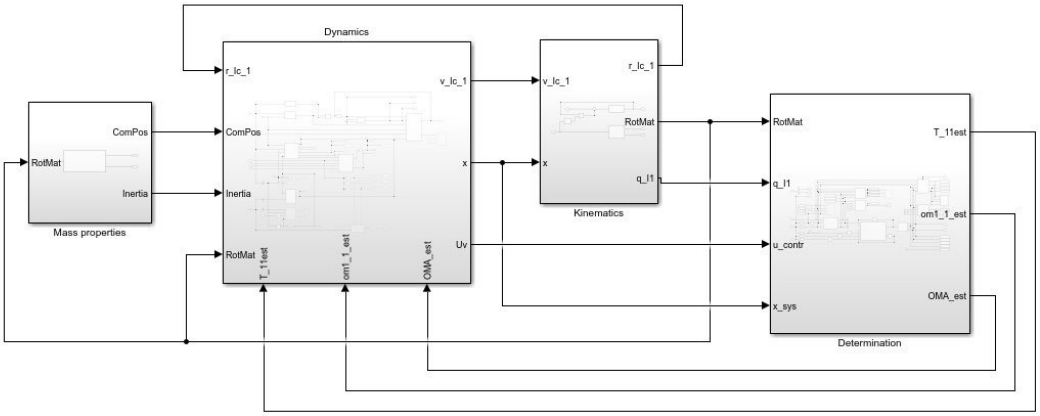

Figure 4.4: Observation on the loop

4.3 EKF results

In this section, the results obtained using the EKF filter model are described. These results will serve as a reference to verify whether the Unscented filter model yields better or equivalent results.
For the integration of the equations of motion, the 5th-order Dormand-Prince method was used, while Forward Euler integration was employed for discrete-time components. The different simulations that have been carried out are briefly described below:

1. **Nominal simulation**: aimed at evaluating the simulation results over a total time interval of two orbits. The antenna is assumed to be balanced, the reaction wheels are unbalanced (resulting in jitter), and the sensors include both systematic and random errors. The results show the evolution of the Absolute Knowledge Error (AKE) of the three measured states: Euler angles, satellite angular velocities, and antenna angle; and the line-of-sight velocity, defined as the projection of $\mathbf{v}_{I,1}$ along the antenna pointing axis $\hat{e}_{8z}$.

2. **Antenna unbalances**: study of the effect of unbalances in the antenna assembly, specifically by varying the mass and position of body 7.

For both antenna imbalance analyses, two sets of simulations were performed. First, a worst-case simulation was considered, adopting the worst-case combinations of the set of free parameters. Second, a Monte Carlo simulation was conducted to evaluate the system's sensitivity and robustness to random errors.

4.4 Reference results

As reference results for the subsequent work, only those related to the nominal analysis are considered. The comparison between the outcomes obtained from the two analyses will be carried out later.
In particular, the following nominal results will be analyzed:

- Absolute Knowledge Error (AKE)
- Power Spectral Density (PSD)

The simulation was carried out with a total duration of 11,400 seconds, corresponding to approximately two complete orbits.
The results obtained with the Extended Kalman Filter (EKF) were analyzed in terms of Absolute Knowledge Error (AKE) for the main system states. The Euler angles exhibit a maximum absolute error below 70 μrad, with very limited random variations and an overall stable trend. The angular velocity error ω_{xx} remains below 3 μrad/s, which corresponds to approximately 0.27% of the nominal angular velocity of the spacecraft in nadir-pointing mode. The error on the antenna angle ϕ_A remains within ± 10 μrad. Regarding the line-of-sight velocity $losv_{I,1}$, which is the key variable analyzed in this study, the maximum observed error stays within ± 1 m/s.
Further analysis was conducted in the frequency domain by evaluating the Power Spectral Density (PSD) of the error in the line-of-sight velocity. In particular, it was observed that the EKF significantly reduces the high-frequency spectral content: starting from approximately 0.1 Hz, the PSD of the filtered signal decreases compared to the unfiltered one [11].

4.5 Unscented Kalman Filter (UKF)

This parts explains the reasons behind the choice to use the Unscented Kalman Filter (UKF). It also outlines the main differences between the Extended Kalman Filter (EKF) and the UKF. The choise to use the UKF is conect by two elements:

- The first factor is the possibility of improving the accuracy of the results. This is due to the fact that, thanks to the filtering capabilities of the UKF, it is possible to further reduce the noise, potentially allowing for better results than those obtained with the EKF.
- The second element is related to the presence of strongly nonlinear equations of motion in this case. For this reason, as will be explained later, the UKF is a more suitable solution in such scenarios.

4.5.1 UKF introduction

The Unscented Kalman Filter (UKF) is an innovative version of the Kalman Filter, designed to improve the limitations of the linear approach used in the Extended Kalman Filter (EKF). It was introduced in the late 1990s by Simon Julier and Jeffrey Uhlmann, with the goal of improving state estimation in nonlinear dynamic systems without the need to compute complex analytical derivatives.[22].
In real systems, such as autonomous vehicles, mobile robots, or satellites, the relationship between the internal state of the system and the measured observations is often nonlinear. In the Extended Kalman Filter (EKF) is a good solution, but it relies on linearizing the model through the computation of the Jacobian matrix. This process can be inaccurate, especially when the system operates in highly nonlinear regions or when the model is affected by uncertainty. [23].
The Unscented Kalman Filter (UKF) stands out for its fundamentally different approach compared to methods based on linearization, thanks to the introduction of the concept of the Unscented Transform (UT) like representation in Figure 4.5, reproduced from [24].

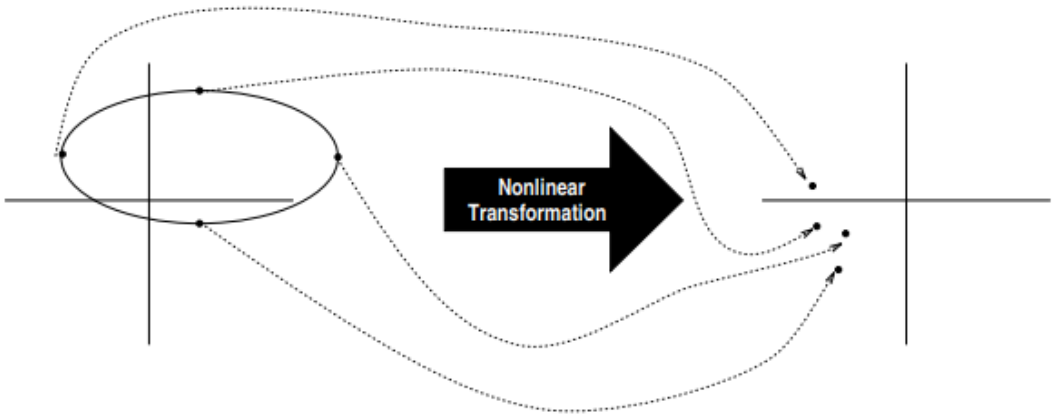

Figure 4.5: The principle of the unscented transform.

This method's main idea is that approximating the probability distribution of the state is more effective and accurate than the nonlinear function that governs the system's evolution. The UKF uses sigma points, a strategically selected set of sample points, to represent the state distribution instead of simplifying the function using a series expansion or computing derivatives. Each sigma point is then propagated through the true nonlinear function of the system, without any approximation or linearization. After the transformation, the resulting sigma points are analyzed to reconstruct a new estimate of the mean and covariance of the state[24] [25]. This process allows the filter to better capture the system's nonlinearity, maintaining high accuracy even when the dynamics are complex or

strongly nonlinear. The main advantage is that the UKF completely avoids calculating the Jacobian, which makes it especially useful in situations where that calculation would be difficult, error-prone, or computationally expensive. As a result, the UKF provides a more stable, accurate, and easier-to-implement solution in many real-world applications compared to the Extended Kalman Filter (EKF).
This approach allows the UKF to achieve more accurate estimates, especially in systems with strong nonlinearities, while avoiding the complexity of calculating Jacobians or Hessians. Moreover, the UKF's structure makes it well-suited for real-time applications, where computational efficiency is essential. Thanks to these features, the UKF has become a reliable alternative to the EKF and is widely used in modern applications such as robotic localization, aerial and space navigation, object tracking, and parameter estimation in industrial processes. Its ability to combine accuracy, robustness, and simplicity has made it a key tool for state estimation in dynamic and uncertain environments [12].

4.5.2 The basic idea

The unscented transformation is a new, novel method for calculating the statistics of a random variable which undergoes a nonlinear transformation. It is founded on the intuition that it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function or transformation. The approach is illustrated in Figure 4.5. A set of points (or sigma points) are chosen so that their sample mean and sample covariance are $\bar{x}$ and P_{xx}. The nonlinear function is applied to each point in turn to yield a cloud of transformed points, and $\bar{y}$ and P_{yy} are the statistics of the transformed points. The samples are not drawn randomly, but rather according to a specific deterministic algorithm. Since there are no issues related to statistical convergence, it is possible to capture higher-order information about the distribution using only a very small number of points [24]. The n-dimensional random variable x with mean $\bar{x}$ and covariance P_{xx} is approximated by $2n+1$ weighted points given by:

$$X_0 = \bar{x} \qquad W_0 = \frac{\kappa}{n+\kappa} \tag{4.14}$$

$$X_i = \bar{x} + \left[\sqrt{(n+\kappa)P_{xx}}\right]_i \qquad W_i = \frac{1}{2(n+\kappa)} \tag{4.15}$$

$$X_{i+n} = \bar{x} - \left[\sqrt{(n+\kappa)P_{xx}}\right]_i \qquad W_{i+n} = \frac{1}{2(n+\kappa)} \tag{4.16}$$

where $\kappa \in \mathbb{R}$, and $\left[\sqrt{(n+\kappa)P_{xx}}\right]_i$ is the i-th column (or row) of the matrix square root of $(n+\kappa)P_{xx}$, and W_i is the weight associated with the i-th point [25]. The transformation procedure is as follows:

1. Instantiate each point through the function to yield the set of transformed sigma points:
$$Y_i = f(X_i)$$

2. The mean is given by the weighted average of the transformed points:
$$\bar{y} = \sum_{i=0}^{2n} W_i Y_i$$

3. The covariance is the weighted outer product of the transformed points:

$$P_{yy} = \sum_{i=0}^{2n} W_i (Y_i - \bar{y})(Y_i - \bar{y})^T$$

The properties of this algorithm are summarized in the following points:

1. Since the mean and covariance of x are accurately captured up to the second order, the calculated values of the mean and covariance of y are also correct to the second order. This means that the mean is computed with a higher order of accuracy compared to the Extended Kalman Filter (EKF), while the covariance is computed with the same level of accuracy. However, there are additional performance benefits. Since the distribution of x is approximated rather than the function $f[\cdot]$, its series expansion is not truncated at a specific order. It has been shown that the unscented algorithm can partially incorporate higher-order information, leading to even greater accuracy.

2. The sigma points capture the same mean and covariance regardless of the choice of matrix square root used. Numerically efficient and stable methods, such as Cholesky decomposition, can be used.

3. The mean and covariance are computed using standard vector and matrix operations. This means that the algorithm is suitable for any process model, and implementation is extremely fast because it is not necessary to compute the Jacobians required by the EKF.

4. The parameter κ provides an additional degree of freedom to "fine-tune" the higher-order moments of the approximation and can be used to reduce overall prediction errors. When $x(k)$ is assumed to be Gaussian, a useful heuristic is to choose $n+\kappa = 3$. If a different distribution is assumed for $x(k)$, a different choice of κ may be more appropriate.

5. Although the parameter κ can be either positive or negative, using a negative value may lead to an invalid covariance estimate (i.e., not positive semi-definite). To avoid this, a modified version of the algorithm can be used, which guarantees a valid covariance matrix. In extreme cases, this version behaves like a second-order Gaussian filter, but without the need to compute complex derivatives such as Jacobians or Hessians [24].

The transformation processes that occur in a Kalman filter consist of the following steps:

- **Predict the new state** of the system $\hat{x}(k+1 \mid k)$ and its associated covariance $P(k+1 \mid k)$. This prediction must take into account the effects of process noise. It involves propagating the current state estimate through the system's dynamic model to anticipate how the state evolves over time.

- **Predict the expected observation** $\hat{z}(k+1 \mid k)$ and the innovation covariance $P_{\nu\nu}(k+1 \mid k)$. This prediction should include the effects of observation noise. By evaluating the observation function on the predicted state, the filter estimates what measurement it expects to receive, and quantifies the uncertainty in this prediction.

- **Predict the cross-correlation matrix** $P_{xz}(k+1 \mid k)$, which captures the statistical relationship between the predicted state and the predicted observation. This term is essential for computing the Kalman gain, which balances the trust between the model forecast and the new measurement [24].

4.5.3 Genric UKF Algorithm

The following description and discussion of the algorithm are based on the following two papers. The first is [25] and [26].

Consider the following nonlinear system, described by the difference equation and the observation model with additive noise:

$$x_k = f(x_{k-1}) + w_{k-1} \tag{4.17}$$

$$z_k = h(x_k) + v_k \tag{4.18}$$

The initial state x_0 is a random vector with known mean $\mu_0 = \mathbb{E}[x_0]$ and covariance $P_0 = \mathbb{E}\left[(x_0 - \mu_0)(x_0 - \mu_0)^T\right]$.

Set Selection of Sigma Points

Let X_{k-1} be a set of $2n+1$ sigma points (where n is the dimension of the state space) and their associated weights:

$$X_{k-1} = \left\{x_{k-1}^j, W_j \mid j = 0, \ldots, 2n\right\} \tag{4.19}$$

Consider the following selection of sigma points, a selection that incorporates higher order information in the selected points:

$$x_{k-1}^0 = x_{k-1}^a \tag{4.20}$$

$$-1 < W_0 < 1 \tag{4.21}$$

$$x_{k-1}^i = x_{k-1}^a + \left[\sqrt{\frac{n}{(1-W_0)}P_{k-1}}\right]_i, \quad \text{for } i = 1, \ldots, n \tag{4.22}$$

$$x_{k-1}^{i+n} = x_{k-1}^a - \left[\sqrt{\frac{n}{(1-W_0)}P_{k-1}}\right]_i, \quad \text{for } i = 1, \ldots, n \tag{4.23}$$

$$\tag{4.24}$$

$$W_j = \frac{1-W_0}{2n}, \quad \text{for } j = 1, \ldots, 2n \tag{4.25}$$

where the weights must satisfy the condition:

$$\sum_{j=0}^{2n} W_j = 1 \tag{4.26}$$

Here, $\left[\sqrt{\frac{n}{(1-W_0)}P_{k-1}}\right]_i$ is the i-th row or column of the matrix square root of $\frac{n}{(1-W_0)}P_{k-1}$. The parameter W_0 controls the position of sigma points: if $W_0 \geq 0$, points tend to move further from the origin; if $W_0 \leq 0$, points tend to be closer to the origin. A more general selection scheme for sigma points, called scaled unscented transformation.

Model Forecast Step

Each sigma point is propagated through the nonlinear process model:

$$x_k^{f,j} = f\left(x_{k-1}^j\right) \tag{4.27}$$

The transformed points are used to compute the mean and covariance of the forecast value of x_k:

$$x_k^f = \sum_{j=0}^{2n} W_j x_k^{f,j} \tag{4.28}$$

$$P_k^f = \sum_{j=0}^{2n} W_j \left(x_k^{f,j} - x_k^f\right)\left(x_k^{f,j} - x_k^f\right)^T + Q_{k-1} \tag{4.29}$$

We propagate then the sigma points through the non-linear observation model:

$$z_k^{f,j} = h\left(x_{k-1}^j\right) \tag{4.30}$$

With the resulting transformed observations, their mean and covariance (innovation covariance) are computed:

$$z_{k-1}^f = \sum_{j=0}^{2n} W_j z_{k-1}^{f,j} \tag{4.31}$$

$$\mathrm{Cov}(z_{k-1}^f) = \sum_{j=0}^{2n} W_j \left(z_{k-1}^{f,j} - z_{k-1}^f\right)\left(z_{k-1}^{f,j} - z_{k-1}^f\right)^T + R_k \tag{4.32}$$

The cross-covariance between x_k^f and z_{k-1}^f is:

$$\mathrm{Cov}(x_k^f, z_{k-1}^f) = \sum_{j=0}^{2n} W_j \left(x_k^{f,j} - x_k^f\right)\left(z_{k-1}^{f,j} - z_{k-1}^f\right)^T \tag{4.33}$$

Data Assimilation Step

We aim to combine the information obtained in the forecast step with the new measured observation z_k. This step, often referred to as the data assimilation or update step, allows the filter to refine its predictions based on real-time data. By incorporating the observation, we can correct the forecast and reduce uncertainty. As in the Kalman Filter, we assume the estimate takes the following form:

$$x_k^a = x_k^f + K_k\left(z_k - z_{k-1}^f\right) \tag{4.34}$$

The gain K_k is given by:

$$K_k = \mathrm{Cov}(x_k^f, z_{k-1}^f)\left[\mathrm{Cov}(z_{k-1}^f)\right]^{-1} \tag{4.35}$$

The posterior covariance is updated using the following formula:

$$P_k = P_k^f - K_k \,\mathrm{Cov}(z_{k-1}^f)\, K_k^T \tag{4.36}$$

Chapter 5

Implementation of UKF

In this section, the implementation process of the new UKF filter within the Determination model will be described. All the attempts made to achieve a correct formulation that produces results consistent with expectations will be detailed. The initial goal of the implementation is to obtain a correct formulation of the filter and to develop a functioning algorithm, both syntactically and conceptually.

5.1 Algorithm implementation

The implementation is divided into sections that reflect the structure of the code. It is important to note that the equations governing the model are the same as those used in the EKF model. The block containing the filter function described in this paragraph is the following 5.1.

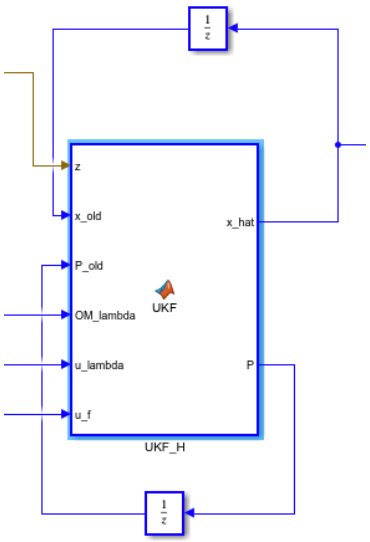

Figure 5.1: UKF block

Initialization

As a first step, it is necessary to define the initial variables of the system. In the EKF, the initial state variable is composed of the following components:

$$\mathbf{x}_k^{f,j} = \begin{bmatrix} \mathbf{q}_{I,1} & \phi_A & \boldsymbol{\omega}_1 & \Omega_A \end{bmatrix}^\top \tag{5.1}$$

where the subscript k indicates the time step, and j refers to the perturbed state. The initial values for the state vector are reported in Table 4.1.
The same applies to the measurement noise matrix $\mathbf{R}$, the process noise matrix $\mathbf{Q}$, and the initial covariance matrix $\mathbf{P_0}$, which use the same values as those in the UKF. The sampling frequency is also the same for both filters; however, in this case, it will be referred to as f_{UKF}.
After initializing the required elements, it is important to identify the reference dimensions of the filter. The filter works with the state vector and the measurement vector $\mathbf{y_k^f}$, so I define:

$$n_{UKF} = dim(\mathbf{x}_0^{f,j}) \tag{5.2}$$

$$m_{UKF} = dim(\mathbf{y}_k^f) \tag{5.3}$$

Actually, m_{UKF} is defined as the dimension of the vector $\mathbf{z}$. The two vectors are equivalent because the vector $\mathbf{z}$ consists of the sensor measurements produced by the sensor block shown in Figure 3.2.
At this point, it is necessary to define the equations that govern the system. These are the same as those used in the EKF model. The system of reference equations, shown in 4.13, was derived in the first chapter.

Definition of sigma point

The next step in the implementation is to define the sigma points. To do this, the matrix $\mathbf{X}$ must be initialized. This matrix contains, as its columns, the state vector perturbed in each of its components. The goal of the sigma points is to define a "cloud" of points around the state vector for each component. For this reason, the matrix $\mathbf{X}$ has dimensions $[n_{\text{UKF}} \times (2n_{\text{UKF}} + 1)]$. The number of columns, $(2n_{\text{UKF}} + 1)$, is due to the fact that the first column corresponds to the unperturbed vector $\mathbf{x}_k^{f,0}$, while the other columns represent positive and negative perturbations of each component of the state vector. As a result, the index j ranges from 0 to $2n$.
In order to define the neighborhood of these points, it is necessary to define the weight W_j, starting with the initialization of the initial weight W_0. This value is arbitrary, but it must satisfy the condition in Equation 4.21. Later, the effects of changing this value will be analyzed. The initial choice was $W_0 = 0$; the closer this value is to zero, the more symmetric the point distribution becomes around the initial point. As a first approximation, the value 0 was selected because it defines a symmetric distribution of points. Using equation 4.25, a vector $\mathbf{W}_{\text{UKF}}$ is built, which contains W_0 as its first element. This vector is mainly used to verify the condition described in 4.26.
Define the matrix before constructing the for loop: **matX**:

$$\mathbf{matX} = Real\left(\sqrt{\frac{n}{(1 - W_0)}\mathbf{P_0}}\right) \tag{5.4}$$

In this way, I obtain a $[9 \times 9]$ matrix that contains the necessary elements to compute all the sigma points. The matrix **matX** is a diagonal matrix which, given the form of $\mathbf{P_0}$, contains the same elements on its diagonal. These elements are called σ. At this point, it is possible to implement the for loop that builds the sigma points matrix:

```
1  X = zeros(n_UKF,2*n_UKF+1);
2  X(:,1) = x_old;
3  matX = real(sqrtm(n_UKF/(1-W_UKF(1))*P_old));
4
5  for i = 2:n_UKF+1
6      X(:,i) = x_old + matX(:,i-1);
7      X(1:4,i) = X(1:4,i)/norm(X(1:4,i));
8      X(:,i+n_UKF) = x_old - matX(:,i-1);
9      X(1:4,i+n_UKF) = X(1:4,i+n_UKF)/norm(X(1:4,i+n_UKF));
10 end
```

Listing 5.1: Sigma point generation in UKF

This *for* loop iterates over the columns of the matrix **matX**, which contain the σ elements, each distributed in the correct position for every column. In the MATLAB code, the state vector at the previous time step is defined as $\mathbf{x}_{\text{old}}$, which corresponds to $\mathbf{x}_0^{f,j}$. It is important to highlight what happens in lines 3 and 4 of the loop shown in 5.1. These lines of code are used to renormalize the quaternion, that is, the first four components of the state vector $\mathbf{x}_{\text{old}}$. A physical quaternion that represents a rotation must have unit norm in order to be valid.

$$\|\mathbf{q}\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \tag{5.5}$$

As a result, a quaternion that does not have unit norm is not a physical quaternion. In other words, it has no physical meaning and cannot be interpreted as a rotation—therefore, it does not represent an attitude measurement. For this reason, every time a quaternion is used in mathematical methods that include operations which may alter its unit-norm property ($|\mathbf{q}| = 1$), it must be renormalized.

$$\mathbf{q}_{\text{normalized}} = \frac{\mathbf{q}}{\|\mathbf{q}\|} = \frac{1}{\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}} \tag{5.6}$$

This renormalization operation is performed for each sigma point of the matrix $\mathbf{X}$. Therefore, the matrix $\mathbf{X}$ is the sigma points matrix constructed as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_{old}^{1,1} & \mathbf{x}_{old}^{1,2} + \sigma & \mathbf{x}_{old}^{1,3} & \cdots & \mathbf{x}_{old}^{1,11} - \sigma & \cdots & \mathbf{x}_{old}^{1,19} \\ \mathbf{x}_{old}^{2,1} & \mathbf{x}_{old}^{2,2} & \mathbf{x}_{old}^{2,3} + \sigma & \cdots & \mathbf{x}_{old}^{2,11} & \cdots & \mathbf{x}_{old}^{2,19} \\ \mathbf{x}_{old}^{3,1} & \mathbf{x}_{old}^{3,2} & \mathbf{x}_{old}^{3,3} & \cdots & \mathbf{x}_{old}^{3,11} & \cdots & \mathbf{x}_{old}^{3,19} \\ \mathbf{x}_{old}^{4,1} & \mathbf{x}_{old}^{4,2} & \mathbf{x}_{old}^{4,3} & \cdots & \mathbf{x}_{old}^{4,11} & \cdots & \mathbf{x}_{old}^{4,19} \\ \mathbf{x}_{old}^{5,1} & \mathbf{x}_{old}^{5,2} & \mathbf{x}_{old}^{5,3} & \cdots & \mathbf{x}_{old}^{5,11} & \cdots & \mathbf{x}_{old}^{5,19} \\ \mathbf{x}_{old}^{6,1} & \mathbf{x}_{old}^{6,2} & \mathbf{x}_{old}^{6,3} & \cdots & \mathbf{x}_{old}^{6,11} & \cdots & \mathbf{x}_{old}^{6,19} \\ \mathbf{x}_{old}^{7,1} & \mathbf{x}_{old}^{7,2} & \mathbf{x}_{old}^{7,3} & \cdots & \mathbf{x}_{old}^{7,11} & \cdots & \mathbf{x}_{old}^{7,19} \\ \mathbf{x}_{old}^{8,1} & \mathbf{x}_{old}^{8,2} & \mathbf{x}_{old}^{8,3} & \cdots & \mathbf{x}_{old}^{8,11} & \cdots & \mathbf{x}_{old}^{8,19} \\ \mathbf{x}_{old}^{9,1} & \mathbf{x}_{old}^{9,2} & \mathbf{x}_{old}^{9,3} & \cdots & \mathbf{x}_{old}^{9,11} & \cdots & \mathbf{x}_{old}^{9,19} - \sigma \end{bmatrix} \tag{5.7}$$

Prediction state

In this phase, the state prediction takes place. Using the dynamic and kinematic equations that govern the system, the state is propagated to obtain its prediction. Obviously, the developed mathematical model has several assumptions that make it ideal:

- Internal friction between components is not included.
- Dissipative effects related to the magnetic dipole are not considered.
- Effects generated by solar radiation pressure are not included.

These effects are neglected because they are of negligible magnitude compared to the others.

```
X_pred = zeros(n_UKF,2*n_UKF+1);
x_pred = zeros(n_UKF,1);
P_pred = zeros(n_UKF);

for i = 1:2*n_UKF+1
    X_pred(:,i) = X(:,i) + tau_filt*f(X(:,i));
    X_pred(1:4,i) = X_pred(1:4,i)/norm(X_pred(1:4,i));
    x_pred = x_pred + W_UKF(i)*X_pred(:,i);
end
x_pred(1:4) = x_pred(1:4)/norm(x_pred(1:4));
for i = 1:2*n_UKF+1
    P_pred = P_pred + W_UKF(i)*(X_pred(:,i)-x_pred)*(X_pred(:,i)-
   x_pred)';
end

P_pred = P_pred + Q;

if min(eig(P_pred)) <= 0
    error('P_pred non positiva');
end
```

Listing 5.2: Prediction state in UKF

In this phase, the prediction of the state at the next time step is calculated using the forward Euler method to integrate the derivative of the state. In the case of the filter, the functions of kinematics and dynamics represented in 4.13 indicate the derivative of the state in each component. The time step depends on the sampling frequency chosen for the filter. In this case, it was decided to keep it constant compared to the EKF, with a value of $f_{UKF} = 10\ [Hz]$ (value reported in Table 4.1).Therefore, the sampling time is equal to:

$$t_{UKF} = \frac{1}{f_{UKF}} \tag{5.8}$$

The two filters have the same sampling time. This choice, as with the previous ones, was made to ensure comparable values between the two filters so that their results can be compared.
In the code syntax, the variable τ_{filt} is used to define the UKF sampling time.
The first step of this phase is to initialize the elements: the predicted state vector $\mathbf{x}_{pred}$, the predicted sigma points matrix $\mathbf{X}_{pred}$, and the predicted covariance matrix $\mathbf{P}_{pred}$.

In the first loop, Equations 4.28 and 4.29 are used to transform the previously calculated sigma points into the predicted state and covariance.
Again, the attitude quaternion is renormalized. The prediction of the state and covariance matrix is done using the weight vector $\mathbf{W}_{UKF}$.
Finally, from line 7 to line 9, an error condition is added on the covariance matrix just calculated.
The matrix $\mathbf{P}_{pred}$ must be at least positive semi-definite, meaning the minimum eigenvalue must be greater than or equal to zero.
The main reasons for these checks are:

- **Represents a real covariance**:
  The covariance matrix describes the variance and correlations between state variables. A non-positive matrix cannot represent a valid covariance.

- **Square root for generating sigma points**:
  In the UKF filter, the square root of the covariance matrix is used to calculate the sigma points.
  This operation requires the matrix to be at least positive semi-definite. Moreover, for this reason, when calculating sigma points—especially when developing the matrix **matX** (Listing 5.1) the function *real* is used.
  This function is necessary because even if the matrix $\mathbf{P}$ is positive definite, there can be very small numerical residues that cause complex parts in **matX** elements on the order of machine precision. Obviously, these elements are not truly complex since their imaginary part is negligible; therefore, the *real* function is used to fix this issue.

- **Numerical stability and filter correctness**: The presence of non-positive matrices can indicate numerical instability.

Measurement prediction

This phase is very important because it involves predicting the measurement based on the measurement function $h(x_k^{f,j})$.
Defining the measurement function is a complex process. The goal is to build a function $h(x_k^{f,j})$ that, given the state vector as input, returns the measured vector z_k^f. Therefore, what the measurement function must do is return a state vector affected by noise.
However, the noise model must be a prediction model, meaning it is based on noise estimates.
This means there are two models for describing the measurement:

- The first model is used in the sensor block, reconstructing a very precise system that, even with some approximations, is very close to the real one. This model cannot be directly used in the filter because it represents the real behavior of the measurement instrument inside the simulator.

- The second model, present in the UKF, tries to reconstruct the correct noise value related to the measurement using estimates not directly linked to the real sensor characteristics (like those shown in Tables 3.1, 3.2, and 3.3).

Building these models is very complex. Also, while they do not affect the filter's correct functioning, they influence the filter's ability to achieve better results.
For these reasons, in the first approximation, the measurement function is considered to

be the identity function, meaning it assumes the predicted state is equal to the measured one.

```
Z = zeros(m_UKF,2*n_UKF+1);
z_pred = zeros(m_UKF,1);
S_zz = zeros(m_UKF);
S_xz = zeros(n_UKF,m_UKF);
for i = 1:2*n_UKF+1
    Z(:,i) = H*X_pred(:,i);
    Z(1:4,i) = Z(1:4,i)/norm(Z(1:4,i));
    z_pred = z_pred + W_UKF(i)*Z(:,i);
end
z_pred(1:4) = z_pred(1:4)/norm(z_pred(1:4));
for i = 1:2*n_UKF+1
    S_zz = S_zz + W_UKF(i)*(Z(:,i)-z_pred)*(Z(:,i)-z_pred)';
    S_xz = S_xz + W_UKF(i)*(X_pred(:,i)-x_pred)*(Z(:,i)-z_pred)';
end

S_zz = S_zz + R;

if min(eig(S_zz)) <= 0
    error('S_zz non positiva');
end

if min(eig(S_xz)) <= 0
    error('S_xz non positiva');
end
```

Listing 5.3: Prediction mesurement in UKF

This approach leads to defining a matrix $\mathbf{H}$ equal to the one used for the EKF, that is:

$$\mathbf{H} = [\mathbf{E_8} \;\; \mathbf{0}_{8\times 1}] \tag{5.9}$$

It is important to point out that this is a $[8 \times 9]$ matrix because the number of measured components in the system is 8, not 9: the satellite's angular velocities around its axes, the attitude quaternion, and the antenna's rotation angle. The antenna's angular velocity is not measured directly but is computed using the dynamic equations that govern the system.
After initializing the main elements the matrix $\mathbf{Z}$, which contains the measurement vector at the current time, and the vector $\mathbf{z}_{pred}$, which represents the predicted measurement vector at the next time step two more matrices are initialized: the cross-covariance matrices $\mathbf{S}_{xz}$ and $\mathbf{S}_{zz}$.
These matrices represent: $\mathbf{S}_{xz}$ the correlation between the variations of the predicted state and the predicted measurement and $\mathbf{S}_{zz}$ the correlation between the actual and predicted measurements. This part of the code is based on the following Equations 4.31,4.32 and 4.33. Also in this case, the predicted measurement quaternion that is, the first four components of $\mathbf{z}_{pred}$ must be re-normalized, because it represents a real measurement quaternion and therefore must have unit norm.
Finally, as with the previously computed covariance matrix, it is also verified here that the two cross-covariance matrices are at least semi-definite positive.

Update

The update step is the final part of the algorithm. In this phase, the new covariance matrix and the new state vector are computed. These new values are calculated using the Kalman gain **K**, which is obtained as the ratio between the two cross-covariance matrices.

```
K = S_xz/S_zz;
x_hat = x_pred + K*(z-z_pred);
P = P_pred - K*S_zz*K';
x_hat(1:4) = x_hat(1:4)/norm(x_hat(1:4));
```

Listing 5.4: Update phase in UKF

Also in this step, the quaternion is renormalized to obtain the final state vector $\hat{\mathbf{x}}$, which will be used for the next iteration. The values that are included in the iteration loop are **P** and $\hat{\mathbf{x}}$. The complete code is shown in Appendix C.
To obtain this code, several iterations were needed to identify both implementation and logical errors. The development process of the algorithm was based on using the results from the EKF as a reference and trying, through multiple attempts, to build a model that produces consistent results.

5.2 Analysis of the identified errors

Before describing the main issues encountered, the following table summarizes the reference result values obtained from the EKF implementation.

Table 5.1: Referement results EKF

Parameter	Value
$\Delta v_{los}\ AKE$	$-0.7 < \ ... \ < -0.1\ [m/s]$
ψ	$10^{-5}\ [rad]$
θ	$10^{-5}\ [rad]$
ϕ	$10^{-5}\ [rad]$
ω_x	$10^{-6}\ [rad/s]$
ω_y	$10^{-6}\ [rad/s]$
ω_z	$10^{-6}\ [rad/s]$
φ_A	$10^{-6}\ [rad]$

During the initial testing phases, the total simulation time is set to 100 $[s]$. A short time is used to allow faster verification of results. Since a correct model is not yet available, running simulations over one or two orbits is not useful instead. At this moment, the priority is to first develop a working algorithm. The reference results of the EKF filter are also calculated with a simulation time of 100 $[s]$, to ensure consistency between the two results. For this design and debugging phase, it is necessary to use the observer out of the loop model. The first results obtained after the implementation of the algorithm are shown in Figures 5.2 and 5.3.

Figure 5.2 shows the results of the AKE of the velocity along the line of sight, which is the main quantity analyzed in this study. To calculate this variable, the true and estimated values of $\mathbf{q}_{I,1}$ and ϕ_A are used to transform the antenna pointing axis $\hat{e}_z^8$ and the orbital velocity vector $\mathbf{v}_{I,1}$ into the same reference frame [11].

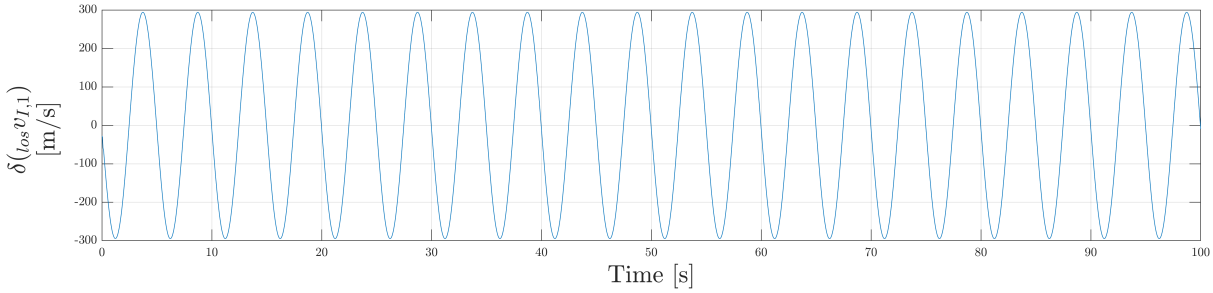

Figure 5.2: Velocity vector component on boresight axis AKE result for the first iteration

Then, the component of the velocity along the real and estimated pointing axes is subtracted to obtain the AKE signal. As can be seen, the results obtained are far from the expected ones. According to Table 5.1, the value of $\delta({}_{\text{los}}v_{I,1})$ is about three orders of magnitude more than what it should be. This suggests that there may be syntax errors in the code or logical errors in the algorithm's structure.

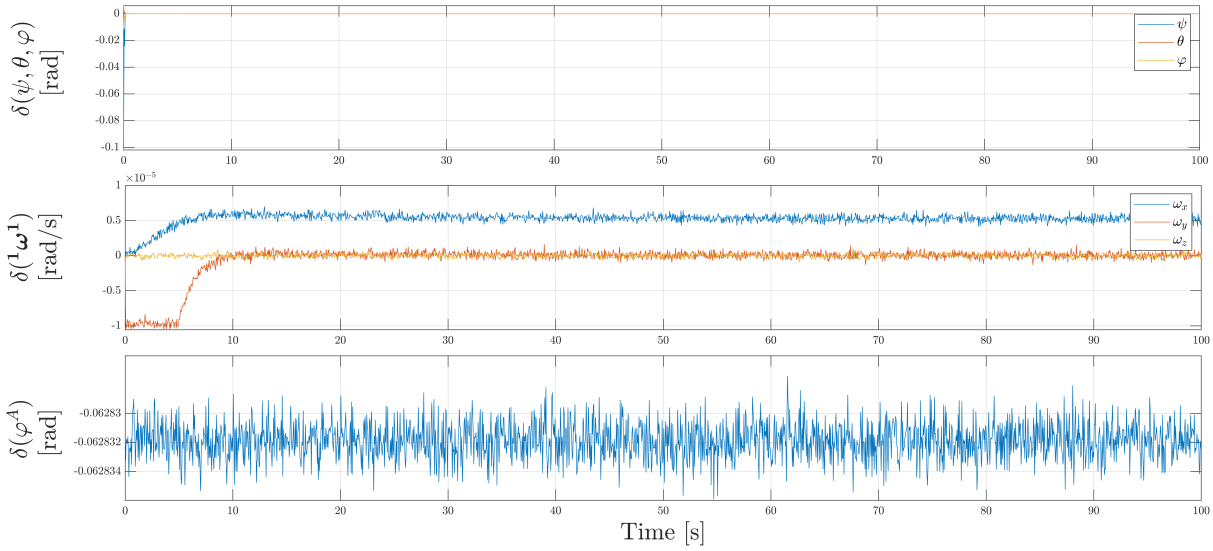

Figure 5.3: Absolute Knowledge Error for the first iteration

Figure 5.3 shows the outputs related to the states: attitude (expressed in terms of Euler angles), angular velocity $\boldsymbol{\omega}_1$, and antenna angle ϕ_A. Again, the values are very large, especially for the Euler angles and the antenna angle ϕ_A. To find a solution to this issue, the first attempt was to simplify the model. One of the first steps is to consider a sensor noise model with only white noise. Although this may seem unimportant, it helps understand whether the problem comes from the filter itself. This operation is performed because reducing the overall complexity of the model helps exclude issues such as strong non-linearities, cross-dependencies, or numerical errors that may indirectly affect the filter and generate errors not directly related to its implementation.
If a simplified measurement model is passed to the filter, and the results remain the same as those in Figures 5.2 and 5.3, then the issue lies in the filter algorithm. A way to investigate the problem is to use a different approach: implementing the model using Euler angles. While the quaternion-based model is more accurate, it is also more complex, increasing the risk of errors that can lead to incorrect filter behavior. To implement the Euler angles formulation, several changes to the model are needed. First, the star tracker

model in the sensor block must be modified. Since Euler angles are a physical vector made up of angles and not a quaternion (which is a physical element made of a vector and a scalar and must have unitary norm), the measurement error can be modeled additively. The model that describes the measurement noise on Euler angles is:

$$\mathbf{\Phi}_{meas} = \mathbf{\Phi}_{true} + \epsilon_{std} + \epsilon_{sys} \tag{5.10}$$

In Equation 5.10, the elements represent the following:

- $\mathbf{\Phi}_{meas}$: the vector of measured Euler angles, $\mathbf{\Phi}_{meas} = [\phi \ \theta \ \psi]^T$.
- $\mathbf{\Phi}_{true}$: the vector of true Euler angles.
- ϵ_{std}: this term represents the random error, i.e., white noise. It follows a Gaussian distribution with a standard deviation of $\sigma = 10^{-6}$.
- ϵ_{sys}: this component represents the systematic error. It includes the worst-case end-of-life systematic bias of the star tracker $bias_{ST}^{EOL}$ and the worst-case thermal effect bias $bias_{TE}$.

$$\epsilon_{sys} = bias_{ST}^{EOL} + bias_{TE} \tag{5.11}$$

To implement this model within the simulators, it is necessary to convert from quaternions to Euler angles (see Figure 5.4).

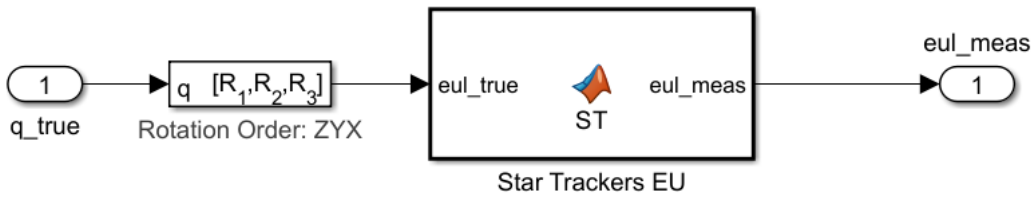

Figure 5.4: Startracker model with a Eulero angle model

This is necessary because the AOCS model outputs quaternions. Therefore, it is necessary to use a block that converts quaternions to rotation angles. Conversely, at the filter output, a block is needed to perform the inverse operation. This must happen because all outputs are built using quaternions. Once a result is obtained, switching between interpretations does not change the results, so it is preferred to output quaternions from the filter to avoid modifying the part of the simulator that generates the outputs. To complete the implementation with Euler angles, it is necessary to reformulate the kinematic equation that governs the prediction model. The state vector changes in its first 4 components. In fact, the state vector $\mathbf{x}_k^{f,j}$ changes its dimension from 9 components to 8. The angular velocity $\boldsymbol{\omega}$ can be written in matrix form as:

$$\begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix}}_{\mathbf{B}} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \tag{5.12}$$

Inverting the B matrix, one obtains the law of evolution of Euler's angles as a function of angular velocity components in body axis, that is:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \frac{1}{\cos(\theta)} \underbrace{\begin{bmatrix} \cos(\theta) & \sin(\phi)\sin(\theta) & \cos(\phi)\sin(\theta) \\ 0 & \cos(\phi)\cos(\theta) & -\sin(\phi)\cos(\theta) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}}_{\mathbf{B}^{-1}} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{5.13}$$

This formulation clearly shows that the derivatives of the Euler angles are a linear combination of the angular velocity components, scaled by the factor $\frac{1}{\cos(\theta)}$, which highlights a singularity when $\theta = \pm\frac{\pi}{2}$ (gimbal lock). In these conditions, the matrix becomes ill-conditioned and the Euler angle representation can lose physical meaning. For this reason, quaternions are used because they avoid this singularity. In this case, however, it is necessary to develop a formulation that can handle the singularity without using quaternions. Special Case for $\theta = \frac{\pi}{2}$(Pitch angle equals 90°):

$$L_{BI}\left(\theta = \frac{\pi}{2}\right) = \begin{bmatrix} 0 & 0 & -1 \\ \sin(\psi - \phi) & \cos(\psi - \phi) & 0 \\ \cos(\psi - \phi) & -\sin(\psi - \phi) & 0 \end{bmatrix} \tag{5.14}$$

When $\theta = \pm\frac{\pi}{2}$, the coordinate transformation matrix does not depend on ψ and ϕ separately, but only on their sum. The code used to implement the new dynamics model in the filter is reported in Appendix D. It is important to emphasize that the kinematics model described with Euler angles was implemented only in the Determination part, while the AOCS model remains unchanged as previously described. Unfortunately, even with this implementation model, the solution does not change; the results remain far from what was expected. However, there is some small improvement regarding the AKE states. In this case, a reduction in the values of the Euler angles components and angular velocities can be seen. The values are still far from what was expected, but an improvement. What also emerges in this case is that the value of the antenna rotation angle remains around 0.063 radians. This was also observed in the quaternion implementation, where the value oscillated around this specific value. If we take the initial value of φ_A, it is equal to 1.2566. The value emerging from the graph is exactly half of what would be expected at time t_0. This deviation from the initial value seems to show that the value taken as input by the algorithm is actually from the previous time step. This would create an imbalance that causes the values of Δv_{los} to be very different from the reference ones.
After noticing this inconsistency, it was found that in the state prediction phase, at line 8 in Listing 5.2, the variables from the matrix $\mathbf{X}$ were used instead of those from $\mathbf{X}_{pred}$. This means that to describe the predicted state, the current state matrix was used instead of the predicted states matrix.
After this correction, the results are much closer to what was expected from the simulator and are shown in Figures 5.6 and 5.7. As can be observed, the value of the antenna rotation angle remains close to zero at this point. This confirms that the filtering process is able to provide a precise and stable estimate for this key parameter. The objective was to achieve a variation in the antenna angle on the order of μrad, and this goal has been successfully met in this latest configuration. This is a crucial requirement from the mission specifications, particularly because even minimal deviations in the antenna pointing direction could negatively impact the Doppler measurements and the effectiveness of communication or observation tasks.

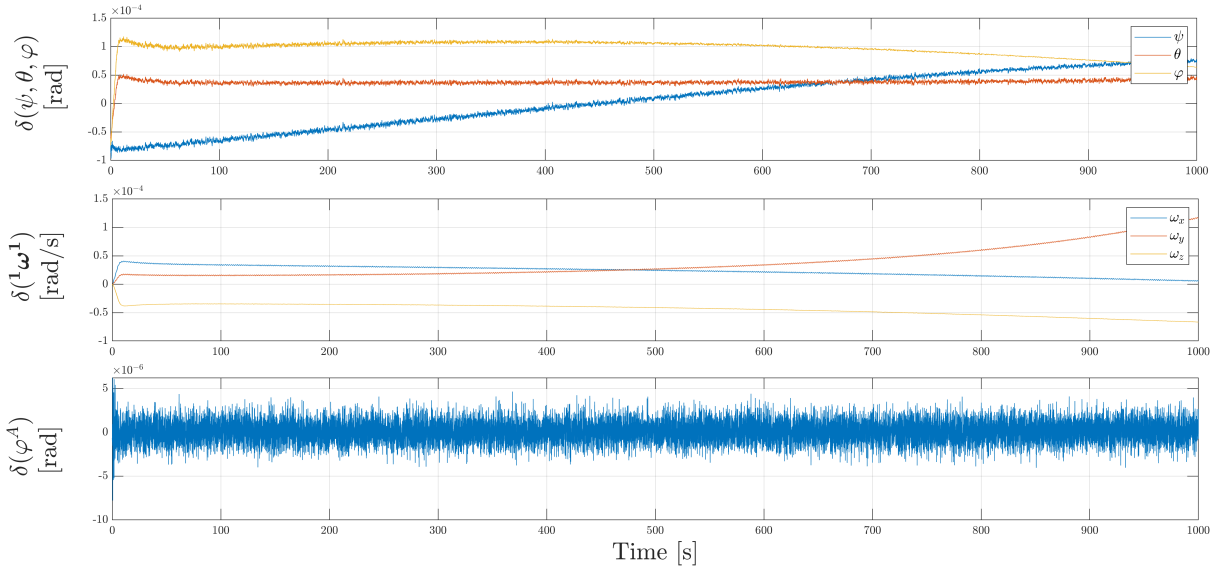

Figure 5.5: AKE state results with Euler's angle model after correction

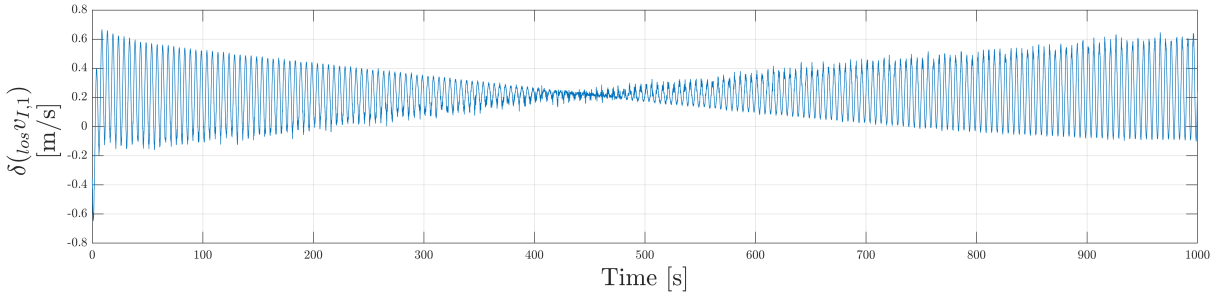

Figure 5.6: Velocity vector component on boresight axis AKE result with Euler's angle model after correction

Despite this improvement, the main remaining issues are related to the divergence effects caused by the yaw angle, denoted by ψ, and by the components of the satellite's angular velocity. These divergence effects manifest as slow drifts or growing oscillations in certain estimated parameters, especially during long-term simulations. Such behavior, while not extreme, could compromise long duration missions or accumulate into significant pointing errors over time. The source of this behavior is most likely related to the way singularities are handled in the Euler angle based model. Specifically, when using Euler angles to represent attitude, certain rotational configurations (e.g., gimbal lock conditions) introduce mathematical singularities that lead to numerical instability in the filter.
The results obtained here represent a substantial improvement compared to those shown in Figures 5.2 and 5.3, where more evident deviations and bias effects were present. To further enhance the quality of the estimation, one possible strategy is to return to the quaternion-based representation of attitude. Quaternions avoid the singularities inherent in Euler angles and offer a more robust and stable solution, especially for systems with continuous or large angle rotations. This alternative formulation can help eliminate the divergence patterns visible in Figure 5.2, ensuring more consistent filter behavior.
The new results obtained with the updated approach are illustrated in Figures 5.8 and 5.7, where a marked reduction in estimation error can be seen. What emerges in this case, however, is that for some values of the state AKE, such as the angular velocities, the

requirements are met, but one component is significantly larger than the others, reaching values of $\boldsymbol{\omega}_x$ around 20 μrad. It is possible that, through further adjustments to the filter, this value could be further reduced. It's worth noting that identifying the root of the issue was not straightforward. The source of the error was deeply embedded in the interaction between the dynamic model and the estimation process, and it required a combination of diagnostic simulations and careful observation of long-term trends to isolate.
In the next chapter, these improved results will be analyzed in greater detail. A more structured tuning process will be applied to refine the filter parameters, such as the noise covariance matrices and the initial state uncertainty, in order to obtain an even more accurate and robust solution. This iterative refinement will aim to ensure that the system not only performs well in nominal conditions but is also resilient to potential disturbances and model inaccuracies.

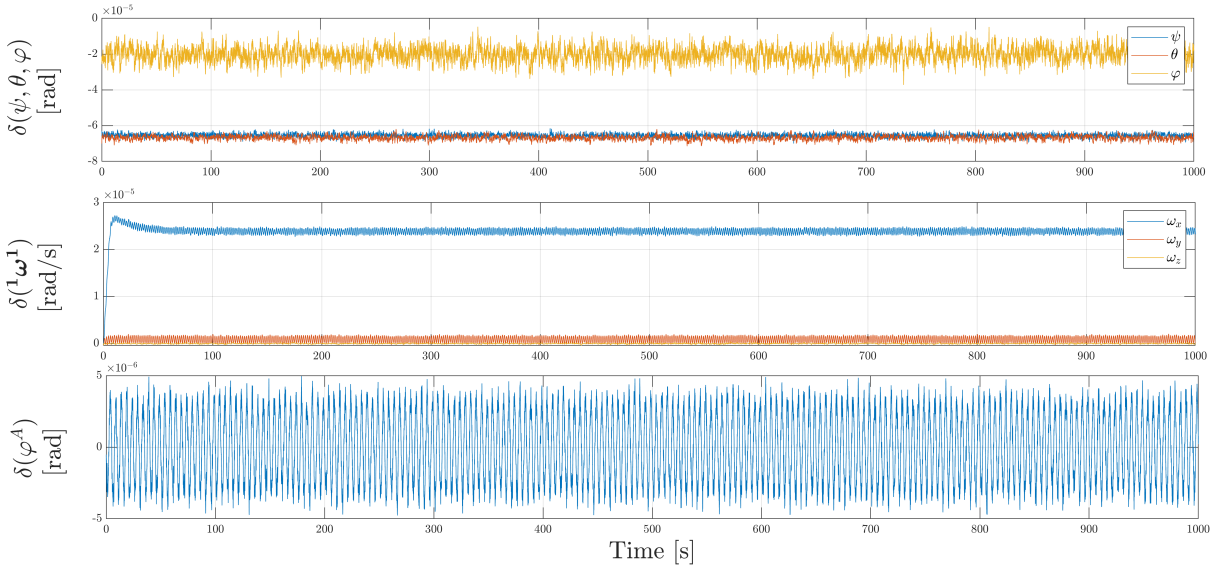

Figure 5.7: AKE state with quaternion model

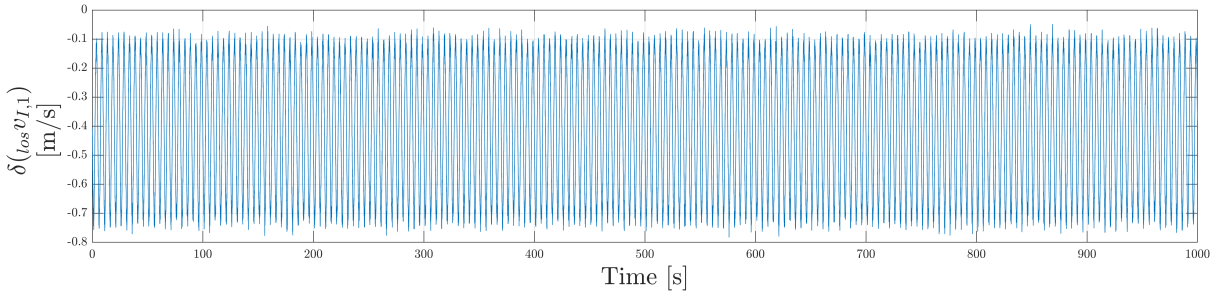

Figure 5.8: Velocity vector component on boresight axis AKE result with quaternion model

Chapter 6

Results

In this chapter of the thesis, the results obtained following the tuning operations will be presented. During the development of estimation algorithms, such as Kalman filters, a fundamental step is parameter tuning. This process involves appropriately adjusting key system parameters such as the covariance matrices of process and measurement noise—in order to improve the accuracy of the estimated states. Tuning allows for a proper balance between the mathematical model describing the system and the sensor measurements. By refining these parameters, it is possible to obtain more stable, precise, and reliable estimates that better reflect the actual behavior of the system. This step is particularly useful in the early stages of design and analysis, when the goal is to better understand the algorithm's performance. Tuning can be performed manually, in an iterative fashion, or by using more advanced automated techniques, depending on the system's complexity and the availability of data. The tuning process is primarily carried out on three key elements: the process noise covariance matrix $\mathbf{Q}$, the measurement noise covariance matrix $\mathbf{R}$, and the initial weight of the sigma points $\mathbf{W}_0$.
As a first step, analyses will be performed using the *Observer in closed loop* model, adopting the same simulation parameters used for the EKF-based model. Specifically, the simulation spans two orbital periods considering that the orbital period is 11400s. As the initial value for W_0, the even value is used 0, a a symmetric distribution.
The simulation parameters are reported in the appendix E and in Table 4.1.
Additionally, this chapter includes simulations related to the antenna unbalance. To study the system response to errors in the balancing of body 7, a set of different simulations was conducted. These simulations involve variations applied to the mass m_7 and to its distance from the hinge point O_A, denoted as $\mathbf{r}_{A,7}$, with respect to the exact values required to achieve perfect static and dynamic balance of the assembly. In particular, worst-case analyses and Monte Carlo campaigns were carried out for different scenarios.
Finally, a comparison between the two models is performed to assess whether improvements have been achieved and to evaluate their impact on the overall solution. The main objective of this work was indeed to determine whether the use of the Unscented Kalman Filter could lead to better results compared to the previously adopted Extended Kalman Filter.

6.1 Filter tuning

As seen in the figures presented in Chapter 5, the filter still requires further adjustments to achieve more efficient calibration. In the process of optimizing the performance of a Kalman filter, an essential phase is the tuning of its key parameters: the process noise covariance matrix ($\mathbf{Q}$), the measurement noise covariance matrix ($\mathbf{R}$), and the initial spread parameter of the sigma points W_0 in the UKF. Proper tuning of these parameters allows the filter to improve its ability to estimate the system state in the presence of noise, disturbances, or incomplete models. A well-calibrated filter is capable of effectively attenuating noise [13].

6.1.1 $\mathbf{W_0}$ sigma point parameters

The parameter $\mathbf{W_0}$ is arbitrary and lies within the range $-1 < \mathbf{W_0} < 1$. This coefficient controls the positioning of the sigma points. $\mathbf{W_0}$ represents the weight associated with the central point of the distribution, which corresponds to the estimated mean of the system. The choice of this weight is crucial: it affects both the spatial arrangement of the other sigma points and the accuracy of the final estimate. The parameter $\mathbf{W_0}$ plays a fundamental role in determining how sensitive the filter is to the nonlinear effects of the system. When the sigma points are placed farther from the mean, they are better able to capture variations caused by nonlinearity, thereby improving the estimation accuracy. However, this configuration may also amplify noise or lead to numerical instability, especially in the presence of complex dynamic models. Additionally, it assigns a specific weight to the central point in the process of reconstructing the mean and covariance matrix after the nonlinear transformation. This weight directly influences the importance the filter assigns to the initial estimate compared to the contributions from the other sigma points. Therefore, an appropriate choice of $\mathbf{W_0}$ is essential to ensure an effective balance between accuracy and robustness of the filter [25] [27]. First of all, it is important to highlight that when positive parameters are chosen, the resulting sigma points generated around the initial point are positioned farther away. Essentially, what happens is that the term inside the square root of equation 4.24 includes the factor:

$$\frac{n}{(1 - W_0)} \tag{6.1}$$

This term tends to infinity as $\mathbf{W}_0$ approaches 1. This means that the closer the value of $\mathbf{W}_0$ is to 1, the farther the sigma points are from the original point, potentially causing a divergence effect in the noise, while yielding very similar solution results. Therefore, it may be preferable to use negative values less than 1. These values allow the sigma points to remain closer to the initial estimate. What emerges from the various simulations is that no significant variation is observed in the results. Consequently, the most appropriate condition is obtained with $\mathbf{W}_0 = 0$.

6.1.2 Q and R matrix covariance process and matrix noise

The objective of modifying $\mathbf{R}$ and $\mathbf{Q}$ is to obtain solutions that are as noise-free as possible, meaning with oscillation amplitudes around the estimated value kept to a minimum.
To achieve this, the approach consists in "unbalancing" the model in favor of the dynamic

prediction, i.e., giving more weight to the model forecast rather than to the measurements. As a first step, it is possible to reduce the matrix $\mathbf{Q}$ by one order of magnitude:

$$\mathbf{Q} = \text{diag}\left(10^{-12} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 100 & 100 & 100 & 1 \end{bmatrix}\right) \tag{6.2}$$

As a first result, it emerges that reducing the covariance matrix leads to a decrease in the amplitude of oscillations for the AKE state values, particularly concerning the satellite's rotational velocities. This indicates a reduction in the measurement noise within the variables themselves, representing a significant improvement in the final outcome. However, a greater oscillation of the Euler angles is also observed; although they remain within a very low order of magnitude, they tend to increase, especially in the φ component. To mitigate this effect, the system can be rebalanced by reducing only certain components of the noise matrix $\mathbf{R}$. Specifically, the noise components related to the quaternion have been reduced by one order of magnitude.

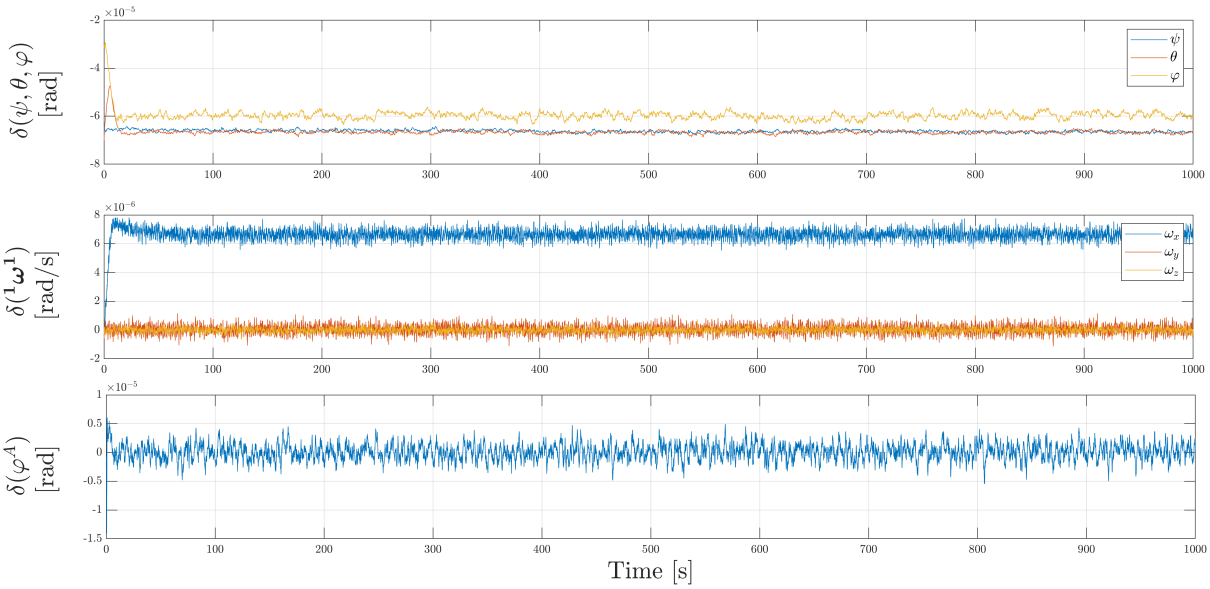

Figure 6.1: AKE state with a different the first four component of noise matrix on time simulation of 1000 s

The resulting noise matrix is:

$$\mathbf{R} = \text{diag}\left(10^{-9} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 0.1 & 0.1 & 0.1 & 1 \end{bmatrix}\right) \tag{6.3}$$

This allowed for a more stable and physically consistent behavior of the Euler angles, improving the overall quality of the simulation. The results are shown in Figure 6.1, where a reduction of one order of magnitude can be observed in the angular velocities, which decrease from 10^{-5} to 10^{-6} rad. This allows compliance with the angular velocity requirements, remaining below the threshold of 100 μrad/s. It is important to emphasize that these results were obtained from simulations run over a shortened time interval, equal to one tenth of the total duration required to cover the evolution of two complete orbits. For this reason, future simulations will be extended to 11,400 seconds in order to realistically represent the long-term dynamic behavior of the system and to obtain results comparable with those reported in [11], where simulations using the EKF filter are carried out over two orbits to assess the satellite's long-term behavior and verify that the mission requirements are met. By performing simulations over two orbits, it can be observed that,

with the current values of $\mathbf{R}, \mathbf{Q}$, and $\mathbf{W}_0$, a variation effect appears in the Euler angles, as shown in Figure 6.2. This effect arises for simulation times longer than one orbit, i.e., after approximately 6000 seconds. The change in the curvature of the solution leads to an undesirable shift in the mean value of Δv along the antenna boresight axis. Specifically, there is a transient phase around 6000 seconds during which the mean line-of-sight velocity drops from an average of 0.4 m/s to a minimum value of about 0.8 m/s, as described in Figure 6.3. This phenomenon is undesirable because the goal is to maintain a stable mean value without large fluctuations. To prevent this effect, further filter tuning is necessary. A possible solution to mitigate this issue is to reduce the noise level on the first component of the satellite's angular velocity. As shown in Figure 6.2, this component exhibits a significantly larger error than the other two, indicating that the Absolute Knowledge Error (AKE) for this parameter is greater. By weighting this component more towards the measurement model, it is possible that the knowledge error will decrease, consequently enhancing the stability of the attitude estimates.
The estimation of the Euler angles is strongly linked to the estimation of the satellite's angular velocity, as the angles are obtained through the time integration of the angular velocity. By modifying the current value of $\mathbf{R}$ and reducing the noise on only the first angular velocity component by one order of magnitude, the filter assigns significantly more confidence to the other two components. This adjustment could help to stabilize the estimation process and reduce long-term drift effects in the simulation.

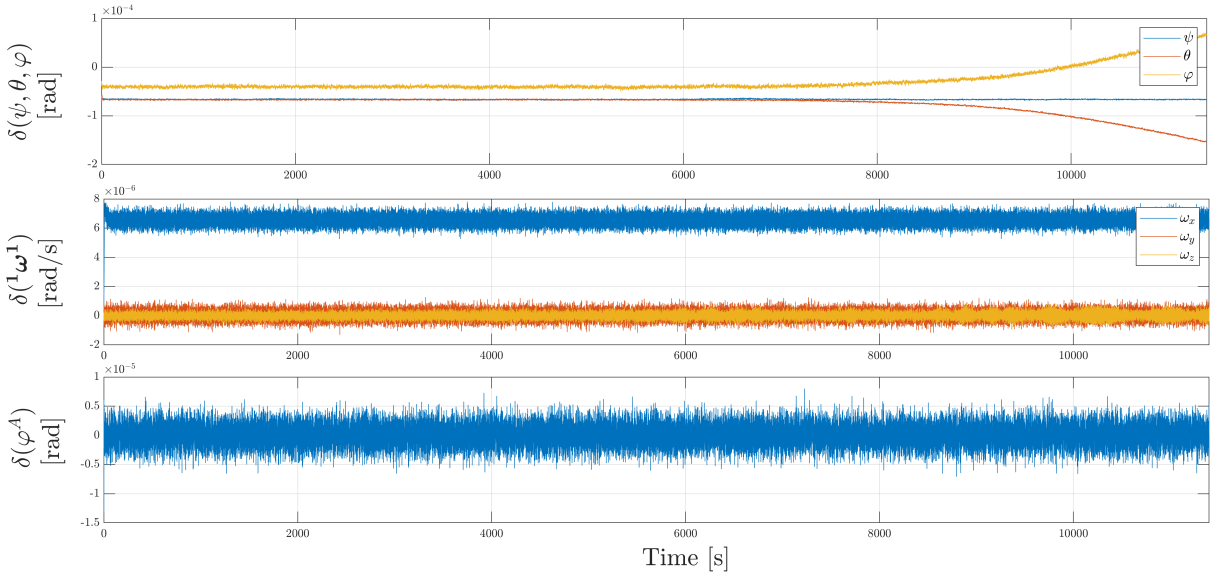

Figure 6.2: AKE state with a time of simulation of two orbit, it's possible to observe a divergence effect around first orbit.

This more strongly constrains the dynamics, preventing the accumulation of the systematic error that previously led to the divergence of the angles. As a result, the noise covariance matrix takes the following form:

$$\mathbf{R} = \text{diag}\left(10^{-9} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 0.1 & 0.1 & 0.01 & 1 \end{bmatrix}\right) \tag{6.4}$$

The final result obtained by performing an analysis with this value of $\mathbf{R}$ leads to improved outcomes in terms of the convergence behavior of the Euler angles. This solution enables an enhanced estimation, which will be presented in the following sections.

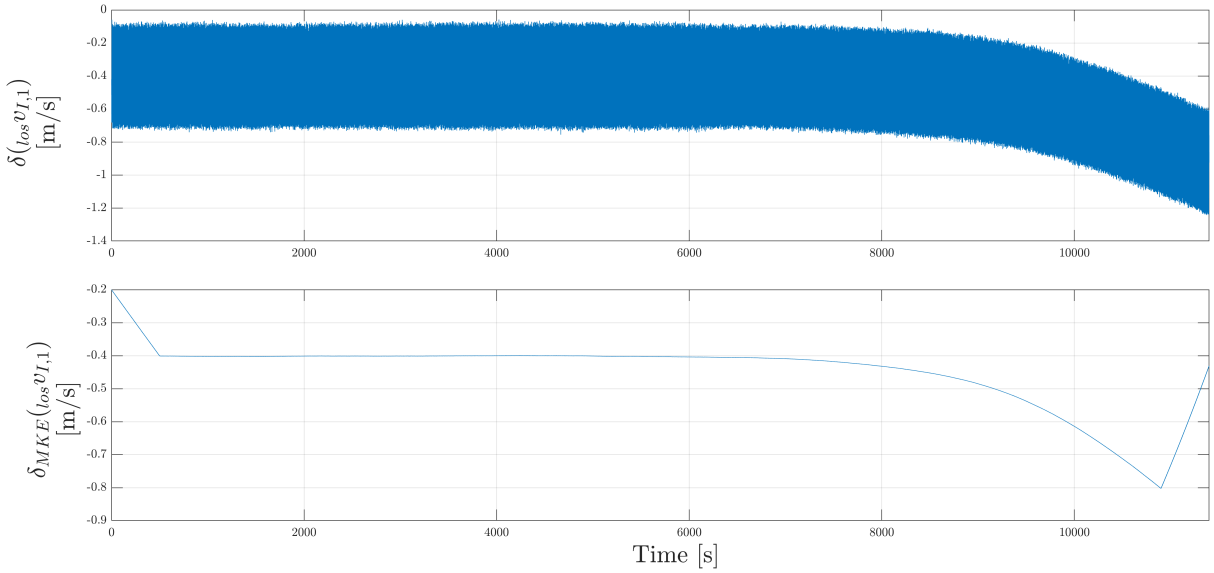

Figure 6.3: Velocity vector component on boresight axis AKE result with a time of simulation of two orbit, it's possible to observe a The change of medium velocity effect around first orbit.

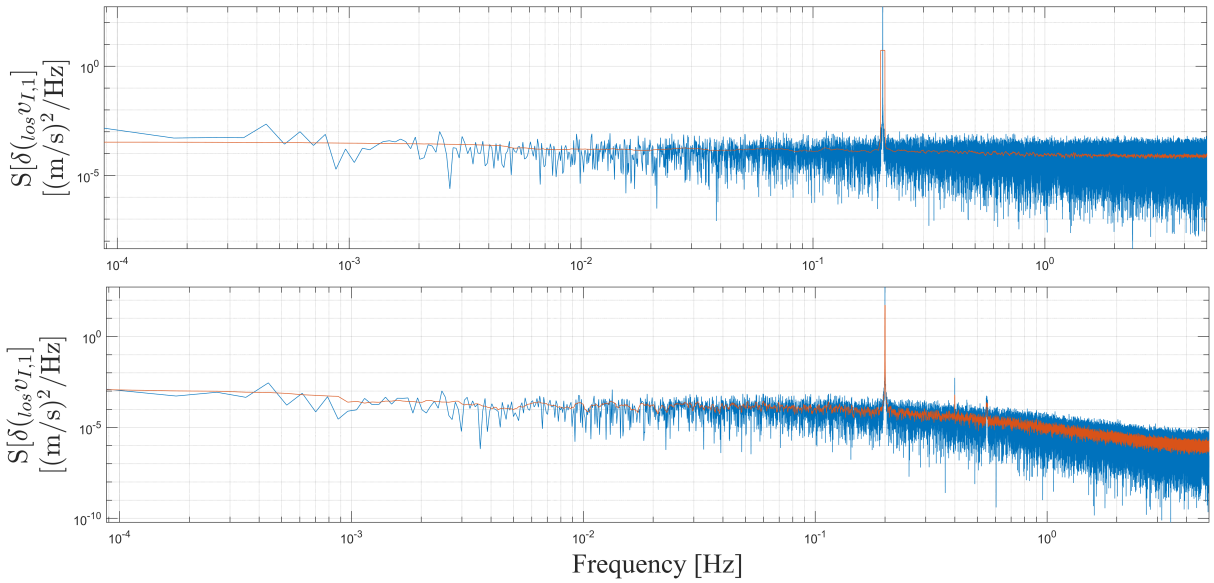

Figure 6.4: PSD of the velocity error Δv_{LOS}: comparison between noisy measurements (top) and filtered estimates (bottom). The red line represents the moving averages, used to smooth the plots and better highlight the overall trends.

The need to fine tune the filter also arises from the analysis of the Power Spectral Density (PSD) of the signal. Figure 6.4 shows two plots representing the PSD of the error signal on the line of sight velocity, denoted as $\delta(_{\text{los}}v_{I,1})$, expressed in $[(\text{m/s})^2/\text{Hz}]$, as a function of frequency (in Hz) on a logarithmic scale. This type of plot allows us to observe which frequency components dominate the error, whether there are specific frequencies where the noise or error is particularly pronounced.
Figure 6.4 is itself divided into two main plots, which together illustrate the functioning of the filter. This plot shows the PSD computed from the unfiltered measurements. The x-axis represents the frequency (in Hz, logarithmic scale), while the y-axis represents the

spectral power expressed in $[(\mathrm{m/s})^2/\mathrm{Hz}]$. The lower plot shows the PSD of the same signal after filtering. The combination of these two plots illustrates how effectively the filter suppresses noise that is, how well the filter performs.
The final PSD trend of the solution after the filtering can be analyzed. As can be observed, the high-frequency PSD decreases much more rapidly and significantly. After the frequency of 10^{-1} [Hz], there is a steady reduction in the PSD value of the velocity. This indicates that the filtering process is functioning correctly.
Furthermore, sharp peaks can be observed at 0.2 Hz, which represents the fundamental rotation frequency of the antenna, and at 0.4 Hz, corresponding to its second harmonic. The latter is presumably due to aerodynamic moments acting on the entire system, which are not modeled in the simplified version adopted. These aerodynamic forces induce a periodic oscillation with twice the rotation frequency, as the cross-sectional area of the system exposed to atmospheric flow varies cyclically with this periodicity.

6.2 Final results

In this section, the final results obtained from a two-orbit simulation will be described. To begin, the main values of the parameters used can be summarized in a table 6.1. Figure 6.5 shows the error related to the line of sight velocity component ($\delta({}_{\mathrm{los}}v_{I,1})$) throughout the simulation, which represents the key result of the study. Once this transformation is completed, the velocity components along the boresight axis both true and estimated are computed, and their difference is evaluated over time.

Parameter	**Value**
$\mathbf{W}_0$	0
$\mathbf{Q}$	$\mathrm{diag}\left(10^{-12} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 100 & 100 & 100 & 1 \end{bmatrix}\right)$
$\mathbf{R}$	$\mathrm{diag}\left(10^{-9} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 0.1 & 0.1 & 0.01 & 1 \end{bmatrix}\right)$

Table 6.1: UKF filter tuning paremeters

The top plot illustrates the instantaneous behavior of $\delta({}_{\mathrm{los}}v_{I,1})$, showing an apparently stationary variation around a stable mean value of approximately -0.4 m/s. As can be seen from the zoomed of 20 s in portion in Figure 6.5, the behavior of the AKE on the velocity oscillates between two limit values, namely -0.1 m/s and -0.7 m/s. This oscillatory behavior is typical when systematic errors are present, such as thermal biases or periodic sensor distortions. When the filter receives input variables affected by low-frequency errors, it tends to produce an oscillatory response directly driven by these phenomena. This result indicates that the absolute knowledge error of the velocity along the line of sight is very low and confined within a narrow range. Therefore, the requirement on the line of sight velocity error, as established by the mission specifications, is met. The lower plot shows the moving average of the error over time. It can be observed that, after an initial transient phase, the average error quickly stabilizes and remains nearly constant for most of the simulation. Only at the temporal boundaries (beginning and end) is a transient in the solution noticeable. This behavior can be attributed to the initial and final effects of the simulation. During these two phases, the filter may have difficulty accurately estimating the values: at the beginning due to possibly inconsistent initial conditions with the physical model, and at the end due to the absence of future data needed for proper averaging.

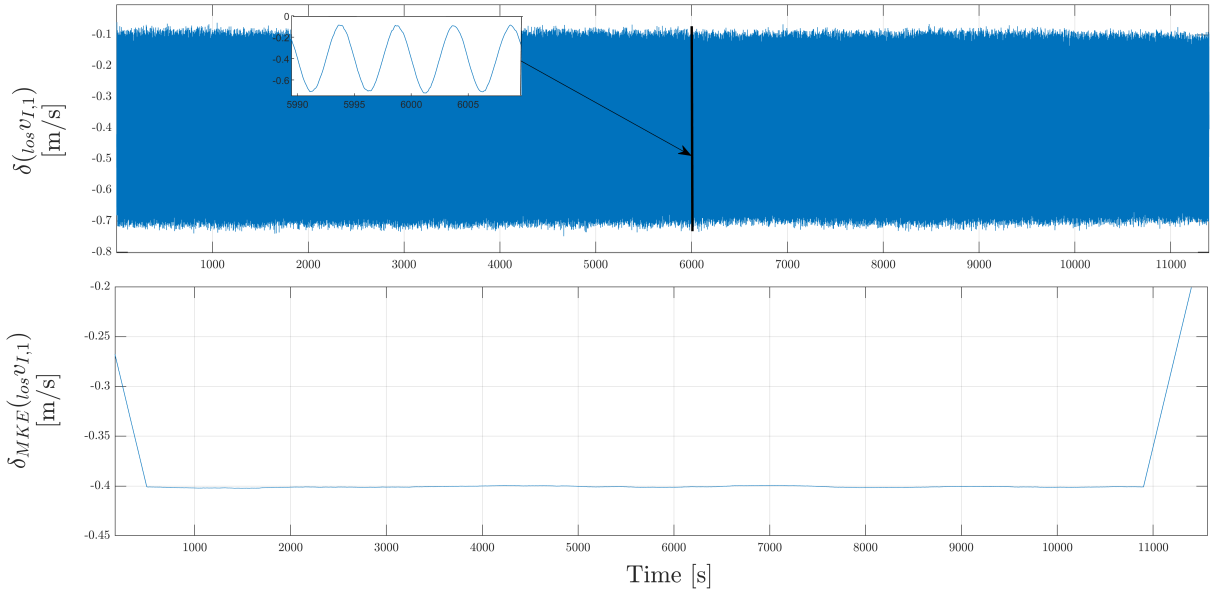

Figure 6.5: Velocity vector component on boresight axis AKE result with a time of simulation of two orbit. In this figure is present a zoom of 20 s to see in details the developed of velocity vector.

Figure 6.6 shows the time evolution of the AKE of the states estimated by the filter throughout the entire simulation. The first panel displays the Euler angles ψ, θ, and φ: their oscillations remain bounded and do not show any divergence, indicating that the system's attitude control is stable. As can be observed, the values consistently remain below 70 μrad.
The second panel refers to the angular velocities along the three principal directions (ω_x, ω_y, ω_z). The signals fluctuate around zero with limited variations, consistent with a regular and controlled dynamic behavior. The AKE on the angular velocities reaches a maximum of approximately 2 μrad/s.
Considering that the constant angular velocity of the satellite in nadir-pointing attitude can be estimated using the LEO orbital period $T = 5700$ s, the orbital angular velocity is:

$$\omega_{orb} = \frac{2\pi}{T} \tag{6.5}$$

This leads to an angular velocity of approximately 0.0011 rad/s, which means that the absolute error on the angular velocity represents only 0.18% of the total angular velocity. Consequently, the error on the angular velocity is very small and has a limited impact.
Finally, the third panel shows the time evolution of the antenna angle, which is also centered around zero and exhibits oscillations within a range of approximately ± 0.5 μrad. The combination of these plots suggests that the filtering and control system is capable of ensuring good dynamic performance and stability over time, even in the presence of disturbances.

6.3 Antenna unbalance

To analyze how the system reacts to possible errors in the balancing body 7, a series of simulations was carried out, each lasting two orbits. As a reference for comparing the effects of such errors, the nominal case described in Section 6.2 was used, considered

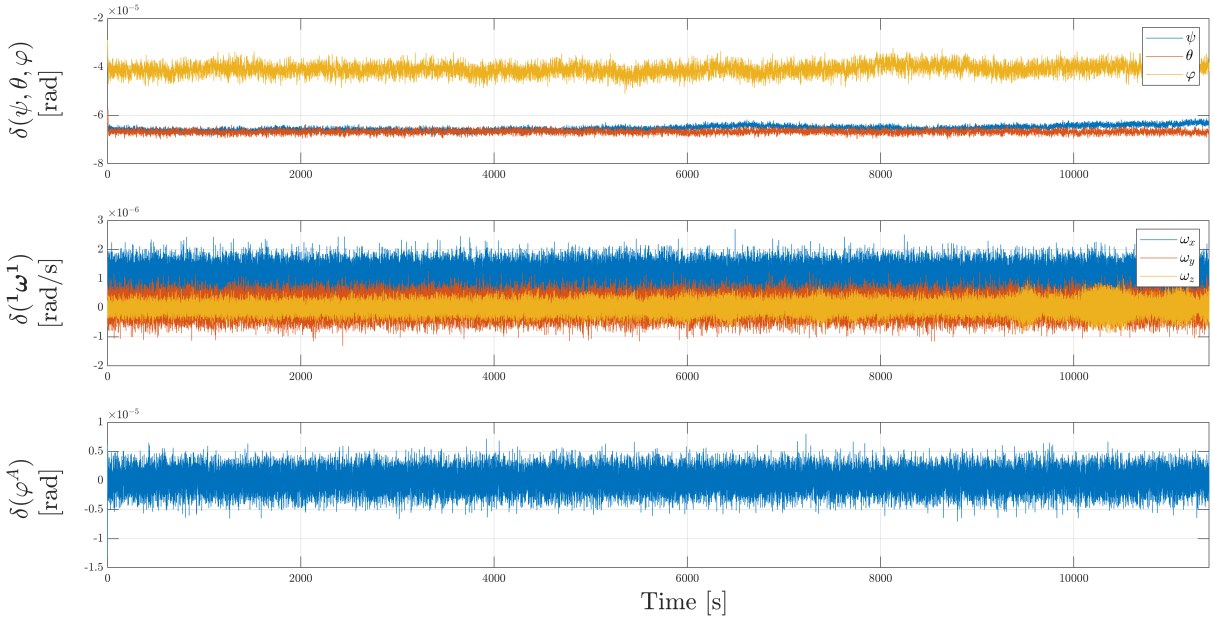

Figure 6.6: AKE state with a time of simulation of two orbit.

as the ideal and disturbance-free scenario. The modifications introduced concern the mass m_7 and the distance from the rotation point O_A, denoted by $r_{A,7}$, with respect to the theoretical optimal values that would ensure perfect static and dynamic balance of subsystem A. Since parameters such as inertia directly depend on these quantities, these properties were also recalculated for each simulated case.
In total, four scenarios were considered, as reported in Table 6.2, in which errors of $\pm\delta m = \pm 10$ grams on the mass and $\pm\delta r = \pm 10$ millimeters on the position were introduced. Although different combinations of these parameters could generate the same unbalancing effect, particularly conservative configurations were chosen, considered representative of plausible errors that might occur during the spacecraft assembly phase.

Simulation	m_7 **error**	$r_{A,7}$ **error**
Case 1	$+\delta_m$	$+\delta_r$
Case 2	$+\delta_m$	$-\delta_r$
Case 3	$-\delta_m$	$+\delta_r$
Case 4	$-\delta_m$	$-\delta_r$

Table 6.2: Error combinations considered for balancing body 7.

From the obtained outputs, it is particularly insightful to analyze the worst-case scenario, which corresponds to the first configuration, namely, the one with the maximum value of mass m_7 and the maximum value of radius $r_{A,7}$. In this case, only the AKE state is considered as output, since the velocity difference between the estimated and actual value along the antenna's line of sight does not show significant deviations compared to the nominal case.
By analyzing the behavior of the Euler angles, as shown in Figure 6.7, it can be observed that the angle φ exhibits a strongly oscillatory trend. This phenomenon is mainly due to the low-frequency error of the star tracker. The irregular pattern of this component suggests the presence of a low-frequency non-white error component, mainly stochastic in nature. Indeed, this effect is expected in the star tracker due to thermal biases related to

the Field of View (FOV), Focal Surface Errors (FSE), and Pixel Sensitivity Errors (PSE). Although the filter performs correctly by keeping the angles below a threshold of approximately $-20\,\mu$rad, it is still important to highlight that the persistent pattern of φ indicates the need to explicitly model and compensate for the sensor's systematic errors. Moreover, the φ component appears to be more affected than the other two, especially if the rotating antenna introduces periodic disturbances that interfere with the accuracy of the star tracker measurement.

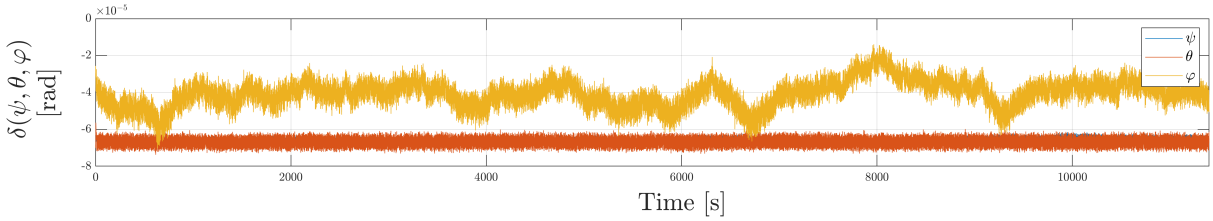

Figure 6.7: Euler angles evolution $\delta(\psi, \theta, \varphi)$ under the worst-case antenna unbalance condition.

Subsequently, it is useful to analyze the trend of the satellite's angular velocities, particularly ω_y and ω_z. From Figure 6.8, it can be observed that the ω_y component exhibits a sinusoidal pattern due to the antenna unbalance dynamics. This periodic behavior is typical of systems where an eccentric mass. In this case, the antenna generates cyclic perturbations on the angular momentum of the main body, the satellite. The antenna imbalance introduces an oscillating disturbance torque along the satellite's transverse axis, which is transmitted through the spacecraft structure, directly affecting the angular velocity around the y-axis. This phenomenon is highlighted in the zoomed-in portion of the plot, where the sinusoidal shape of the ω_y curve is clearly visible, while the ω_z component appears less affected. This effect represents a secondary but significant consequence of the antenna imbalance, which, due to internal couplings within the system, transmits perturbative moments that in high precision systems such as WIVERN could degrade mission performance.

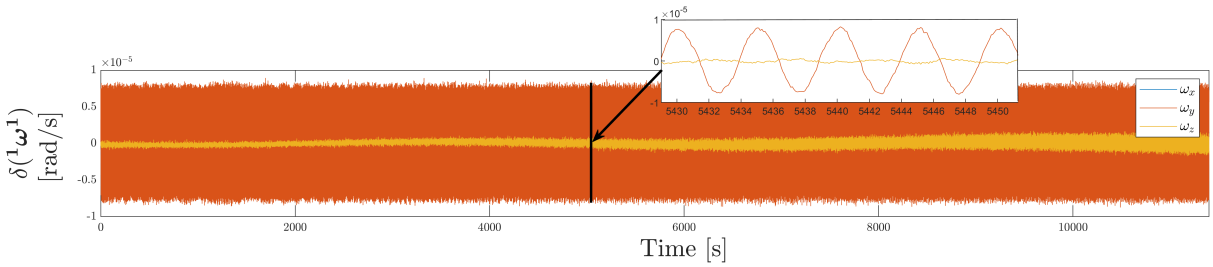

Figure 6.8: Angular velocities of the spacecraft $\delta(\omega_x, \omega_y, \omega_z)$ with zoom of variable trends.

The divergent behavior of ω_z, on the other hand, is attributed to the jitter phenomenon of the reaction wheels. During operation, reaction wheels can generate high-frequency mechanical vibrations that propagate through the satellite's structure. In this case, the divergent trend of ω_z suggests the introduction of a non-negligible perturbation along the z-axis. Notably, over the operational period of two orbits, a variation of one order of magnitude is observed in ω_z, increasing from an average value on the order of $10^{-7}\,\mu$rad to

10^{-6} μrad. Finally, Figure 6.9 shows the evolution of the antenna rotation angle. In this case as well, a slight oscillatory phenomenon can be observed, caused by the imbalance introduced in the antenna. However, the values remain within the limits required by the mission specifications.
Overall, it can be concluded that the filter performs its function correctly, as it keeps the values of the various components below the required thresholds even in the presence of perturbations.

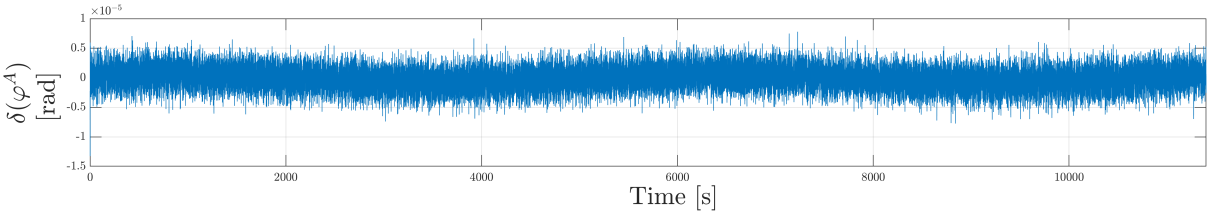

Figure 6.9: Error on the antenna rotation angle $\delta(\varphi^A)$ with respect to its expected position.

In addition to the worst-case analysis, a Monte Carlo analysis was conducted to study the behavior of the velocity error along the line of sight of the antenna. Monte Carlo analyses are typical of complex engineering models and are used to verify the robustness of a solution or model with respect to random variations in input variables. Often, these variables are assumed to follow a Gaussian distribution, in order to gather information about the satellite's actual behavior in the presence of chaotic perturbations. A total of 60 simulations are conducted, in order to obtain a confidence level of 95 %. This allows us to consider that the obtained results cover the actual variability of the system in the presence of the considered uncertainties.
A scatter plot can be constructed, where the x-axis represents the amplitude of the error (the maximum variation) of the velocity along the line of sight in the F_I frame, and the y-axis shows the absolute mean value of the velocity error, representing the constant bias present in the measurement. In the plot, the orange dots represent the 60 Monte Carlo simulations performed by introducing a random uncertainty ($\pm\delta m$ and $\pm\delta r$) on the parameters m_7 and $r_{A,7}$, corresponding to the mass and the center of mass of the antenna, respectively. The blue dots represent the worst-case scenarios, generated by the most unfavorable combinations of the two parameters.
By conducting the analysis over two orbits, as shown in Figure 6.10, two main considerations emerge. First, the system proves to be robust, as all results remain well below the 1 m/s mission requirement. The worst-case analysis reveals amplitudes hovering around values of 0.401 m/s and 0.400 m/s. The Monte Carlo simulations display a moderately scattered distribution of the velocity error amplitude, with values ranging approximately from 0.51 m/s to 0.56 m/s.
However, it is evident that the worst-case values are more conservative than those obtained from the Monte Carlo simulations. Ideally, one would expect the Monte Carlo results to lie within the bounds defined by the worst-case scenarios. This unexpected behavior may be caused by several factors. One possibility is that the filter tuning still requires further refinement, as plotting the individual Monte Carlo results for the AKE of the Euler angles reveals a divergence phenomenon after the first orbit, similar to what is shown in Figure 6.2.
Nonetheless, this phenomenon may also be related to other aspects of the analysis itself, and therefore it cannot be conclusively attributed to the filter tuning alone. Moreover, it

is possible that this phenomenon arises precisely from the nature of the UKF, namely, its ability to better capture the non-linearities present in the system. It can be assumed that this characteristic makes it more sensitive to variations, which in turn leads to greater dispersion.

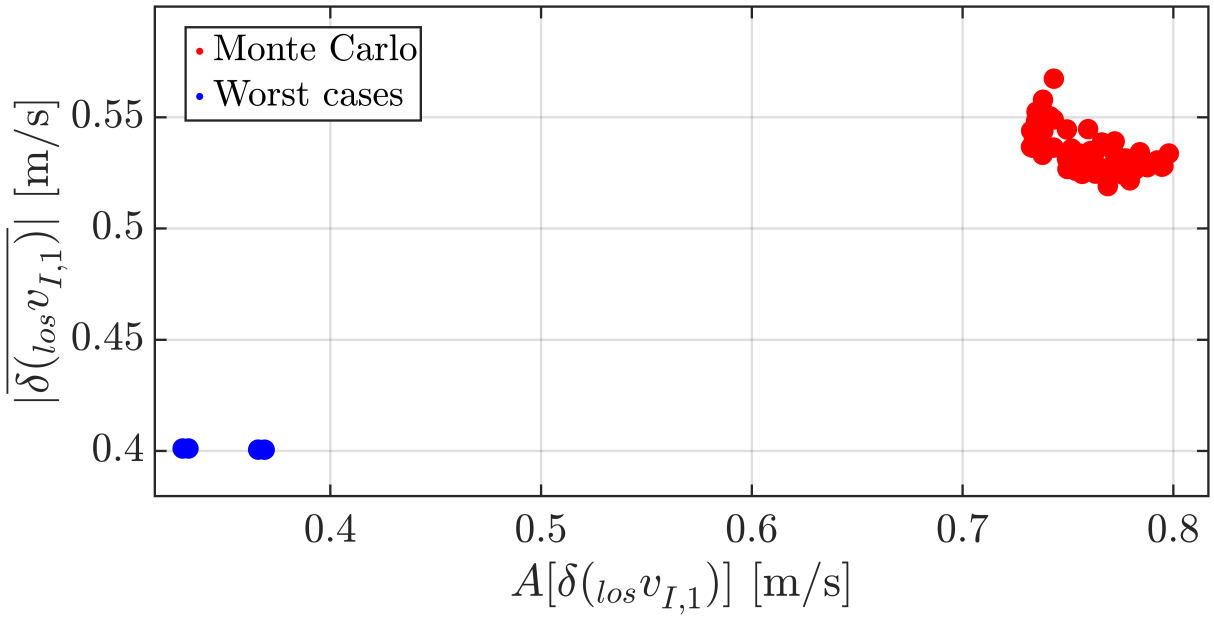

Figure 6.10: Comparison between Monte Carlo simulations (red dots) and worst-case scenarios (blue dots) in terms of velocity error along the line of sight, for a simulation time corresponding to two complete orbits.

Subsequently, an analysis was carried out again using 60 simulations over a simulation time corresponding to one orbit, as shown in Figure 6.11.
This was done to observe the system's behavior after a single orbital period. Compared to the previous case, the values are more contained and lie closer to the worst-case values. The results of the Monte Carlo simulation range between 0.404 m/s and 0.402 m/s. However, even in this case, the values do not fall within the worst-case bounds, although they exhibit a more compact and regular distribution and a decreasing trend. This phenomenon just like in the previous case shown in Figure 6.11 could be the result of the need for further tuning.
It is important to emphasize that finding the optimal filter parameters to achieve consistent results from every point of view is an iterative process, which requires multiple adjustments before identifying the most efficient configuration.
In conclusion, it can be stated that the filter still demonstrates robust performance, especially for simulation times equal to one orbit, but also for two orbits, as the results remain below the required thresholds in both cases.

6.4 Comparison between EKF and UKF

When analyzing the results obtained with both models the one using the EKF and the current model with the UKF very similar outcomes emerge. In both cases, the mission requirements are met. This is a highly positive result, as it allows us to state that both models function correctly and meet the expected performance levels. The main differences

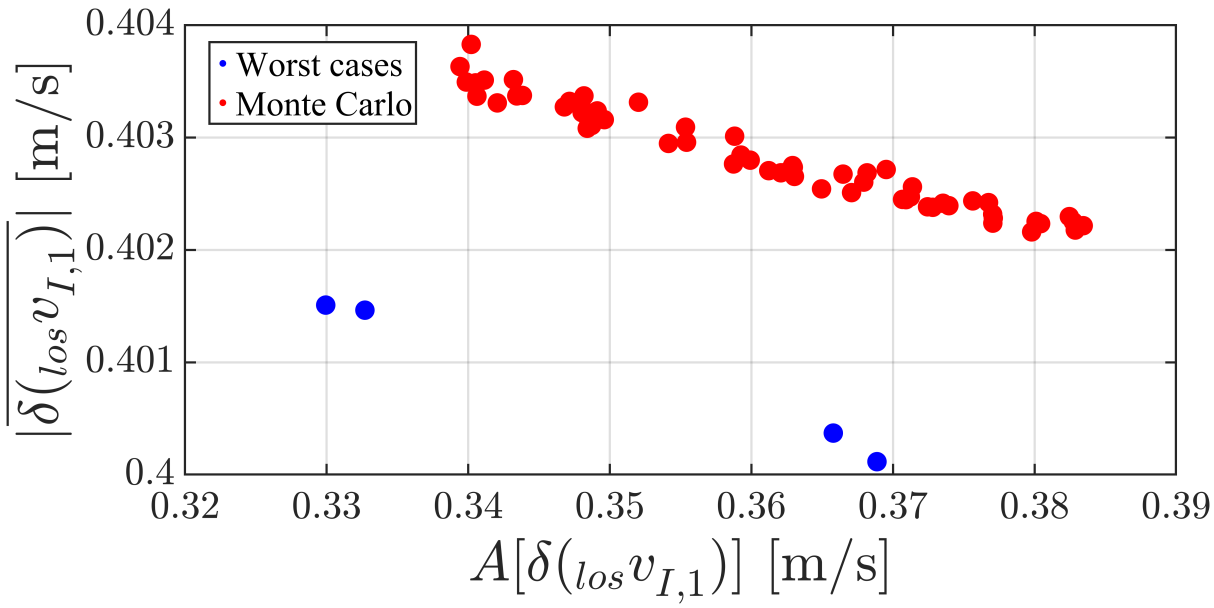

Figure 6.11: Comparison between Monte Carlo simulations (red dots) and worst-case scenarios (blue dots) in terms of velocity error along the line of sight, for a simulation time corresponding to one complete orbit.

between the two models primarily arise in the AKE state, particularly in the values of the Euler angles, and in the tuning parameters. First, referring to the two tables presented in the text, Table 4.1 and Table 6.1, we can observe differences in the values used for matrices **R** and **Q**. What emerges from these parameter values is that the model using the Extended Kalman Filter adopts more conservative parameters compared to the Unscented Kalman Filter. This behavior could be an indication that the UKF performs the filtering process more efficiently, due to its nonlinear nature. Specifically, the following points can be observed:

- In the UKF, as shown in Table 6.1, the values in matrix **Q** are significantly lower, and some components of matrix **R** are also smaller than those used in the EKF. This suggests that the UKF is able to achieve effective estimation while relying more heavily on the dynamic model.
- In the EKF, both **Q** and **R** have higher component values. This may indicate that the EKF requires higher noise levels to compensate for the approximations introduced by the linearization process.

In conclusion, it can be stated that the model with the UKF achieves the same results as the EKF, but with stricter parameters more weighted toward the model. While this is a secondary result, it may be a sign that the UKF offers greater robustness to nonlinearities, better predictive capabilities of the dynamic model, and improved numerical efficiency. This outcome aligns well with expectations, as the UKF is known to perform better when estimating dynamic states in systems with strong nonlinear behavior.
As shown in Table 6.3, which reports the EKF results taken from the reference paper [11], slight differences emerge in the values of the Euler angles and in the first component of the angular velocity. Specifically, the value of ψ changes from -20 μrad to -40 μrad, which could indicate that the UKF provides a more pronounced estimate for this angle compared to the EKF. This result might be attributed to the greater sensitivity of the

UKF when dealing with highly nonlinear dynamic systems. Secondly, the first component of the satellite's angular velocity decreases from a maximum value of 3 μrad/s to 2 μrad/s. The UKF appears to yield a slightly lower estimation error (closer to zero), which could suggest better noise suppression or a more effective adaptation to the dynamic model.

Results	**EKF**	**UKF**	**Unit of measurement**
ψ	-70	-70	μrad
θ	-70	-70	μrad
φ	-20	-40	μrad
${}^{1}\omega_{x}^{1}$	...<-3	...<-2	μrad/s
${}^{1}\omega_{y}^{1}$	-1.5<...<1.5	-1.5<...<1.5	μrad/s
${}^{1}\omega_{z}^{1}$	-1.5<...<1.5	-1.5<...<1.5	μrad/s
φ^{A}	-5<...<5	-5<...<5	μrad/s
$\delta({}_{los}v_{I,1})$	-0.1<...<-0.7	-0.1<...<-0.7	m/s

Table 6.3: Comparison between EKF and UKF results

For the other components, both filters produce equivalent results. In conclusion, the work done provides some promising indications that the UKF may be a more suitable solution for reducing the absolute error component in this specific scenario. Moreover, since the UKF implemented is a basic version, further improvements could be achieved by adopting more sophisticated models for state prediction.
A comparison can then be made between the results obtained from the antenna unbalance analysis. When using a UKF model, the results tend to span a wider range and appear more detached from the worst-case scenarios compared to those presented in the article [11]. In contrast, the EKF yields more compact distributions and a stronger correlation with the worst case data indeed, the results fall within the bounds of these cases.
This behavior reflects one of the key characteristics of the UKF: its ability to handle system nonlinearities more accurately through the unscented transformation. This method allows for a better capture of possible state variations, although at the cost of a broader dispersion in the results.
In conclusion, the EKF proves effective in terms of consistency and compactness of the outcomes, but it may not be sufficiently robust for complex scenarios. The UKF, while showing greater dispersion, provides a more conservative and potentially more reliable representation of the system's behavior, especially in the presence of strong nonlinearities. However, it is important to emphasize that further analysis would be needed to fully understand this phenomenon and determine whether the observed effect is due to the introduction of a more sensitive filter model or instead the result of other factors such as tuning or specific analysis parameters. Based on the data currently available, one can hypothesize that a combination of these factors may be contributing to this outcome, which remains highly interesting, as it appears to highlight a fundamental feature of the UKF.

Chapter 7

Conclusions

This work addressed the problem of implementing an Unscented Kalman Filter (UKF) within a satellite model equipped with a Doppler radar and a rotating antenna. The satellite under consideration exhibits strong nonlinearities due to the complexity of its dynamics and the presence of a rotating payload. The primary objective was to verify whether the use of a UKF could ensure compliance with mission requirements, namely an Absolute Knowledge Error (AKE) below $100, \mu$rad for each axis and an error below $1, \mathrm{m/s}$ for both the main body attitude and the velocity along the Line of Sight (LOS), respectively. Additionally, the study aimed to evaluate whether the UKF could offer superior performance compared to the Extended Kalman Filter (EKF) in terms of absolute estimation accuracy.

To achieve these goals, it was first necessary to investigate the state of the art of the model from both a mathematical and conceptual point of view. Then, the structure of the simulator developed in Simulink was analyzed, with particular focus on both the orbit control subsystem and the attitude determination module. After understanding how the existing models operated, the new UKF filter was designed and defined, both mathematically and in terms of implementation: initially via a MATLAB script, and then by integrating it into the Simulink simulator.

Once implementation was completed, numerical tests and in-depth analyses of the obtained results were carried out. Furthermore, additional simulations were performed to assess the system's behavior in the presence of imbalances generated by the rotating antenna, from both static and dynamic perspectives. The results obtained suggest that the filter functions correctly and allows the system to meet the required mission specifications. On the other hand, it is important to highlight that, compared to the EKF based model, there is greater variability in the results. This effect could also be due to an improvement introduced by the UKF, which allows for a better capture of system non-linearities. Further analyses will be necessary to fully understand this phenomenon.

Ultimately, the results demonstrate that the UKF filter allows mission requirements to be met with a wide safety margin from all perspectives. The difficulties encountered during the filter's development and integration also helped to highlight some structural weaknesses of the system, both in terms of algorithmic implementation and mathematical modeling.

It is important to note that the implemented UKF version is a basic form, in which the measurement function is approximated by an identity matrix, thus simplifying the observation process. Nonetheless, the UKF yields slightly better results, particularly in certain aspects. Consequently, it can be stated that, in some respects, the UKF

filter enhances the performance of the EKF-based model. However, it should also be acknowledged that the EKF has already achieved a very high level of accuracy for the considered case, making the potential benefits of introducing the UKF marginal. Further improvements in estimation accuracy may require the use of more advanced techniques or the optimization of the observation model under more realistic and complex conditions.

7.1 Future works

There are several possible solutions to improve the results obtained, both in terms of modeling and filtering. To enhance the model and make it more consistent with reality, current simplifications could be removed and descriptive models for currently neglected effects could be developed. In particular, one could consider introducing models that simulate, at least partially, unbalance effects, vibrations, or external forces that may generate instability phenomena. Below is a list of possible improvements to be integrated into the system:

- Review the Monte Carlo and Worst-Case analysis to identify potential system improvements and determine the most suitable parameters for more effective tuning.
- Introduce a simulation model for measurement effects caused by temperature gradients.
- Assess the stability against antenna unbalances, as already done, but in a different manner. Specifically, consider variations in the vector $\mathbf{r}_{A,7}$ as three-dimensional rather than one-dimensional.
- Propellant movement in the tanks (sloshing) can cause small forces and moments that affect the satellite's dynamics.
- Vibrations of the solar panels, due to their flexibility, can generate low-frequency moments that compromise the system's stability.
- Solar radiation pressure changes with orbit and can produce accelerations and disturbances that must be considered in the dynamic balance.
- Residual magnetic torque can vary with orbital periodicity and affect attitude if not actively compensated.

Including such phenomena in the model would significantly increase the realism of the simulations, enhance the robustness of the estimation filter, and optimize the overall system performance in a real space environment.
Based on these considerations, it is possible to reflect on additional solutions that could be integrated into the filter model to obtain more accurate results. In particular, two possible implementation approaches can be identified that aim to improve the measurement function and, consequently, the effectiveness of filtering. The first approach could involve the use of more sophisticated probabilistic models to represent noise, while the second could focus on the adaptability of the filter under varying operational conditions. Both methods aim to make the estimation system more accurate, stable, and robust.
In the first case, a possible approach consists of describing the measurement function using noise prediction models based on random or Gaussian distributions. This strategy could allow for more advanced filtering, improving estimation accuracy and ensuring greater

system reliability.
The second approach involves the use of the Augmented Unscented Kalman Filter (AUKF). The Augmented UKF represents an evolution of the classical UKF, designed to handle uncertainty in nonlinear dynamic systems with additive noise more precisely. Unlike the traditional UKF, in which the system state is described only by dynamic variables, the AUKF expands the state by also including process and measurement noise as explicit components of the state vector. This "augmented" approach allows direct handling of disturbances during the generation of sigma points, which are at the core of the unscented method.
The main advantage of this technique is its greater ability to capture the true propagation of uncertainty in the system. In particular, when the noise is moderate or low, the Augmented UKF offers more accurate estimates than the standard UKF. This is because, by including noise terms in the prediction step, it can more realistically model their effects on the system state, without having to treat them as a simple external addition in the covariance. In application scenarios such as object tracking, attitude control, or satellite navigation, this can lead to significant improvements in estimation quality, in terms of lower mean error and greater robustness to measurement noise. For example, if a system has precise sensors but is subject to small disturbances, the AUKF is particularly suitable because it structurally accounts for these perturbations during the state update process. However, it is important to emphasize that the AUKF is not always the best overall choice: when noise levels are very high, incorporating it into the state vector may amplify uncertainty rather than reduce it. In such cases, a standard UKF, which maintains a clearer separation between state and noise, may yield more stable results. In summary, the Augmented Unscented Kalman Filter is a powerful tool for improving estimation accuracy in nonlinear dynamic systems, but its effectiveness strongly depends on the noise profile in the system. When properly calibrated, it can lead to clear improvements over the standard UKF, especially in scenarios where precisely modeling even the smallest uncertainties is crucial [28].

Appendix A

Nomenclature

$\textcircled{i}$ Center of mass of body i.

$\textcircled{\bullet}$ Center of mass of the spacecraft.

m_λ Mass of body λ.

m Mass of the spacecraft.

0^i Origin of the i-th reference frame

$\hat{e}_i^j$ Axis $i = x, y, z$ of frame $\mathcal{F}^j$.

$\mathbf{r}_{i,j}$ Position vector from point O^i to O^j.

$\mathbf{v}_{i,j}$ Absolute velocity vector of point O^j.

$\mathbf{I}_i^{(\lambda)}$ Inertia dyadic of body λ (or the spacecraft, if right superscript is missing) measured about point O^i.

$\mathbf{J}_{ij}^{(a)}$ Mixed inertia dyadic of body λ, measured about points O^i and O^j.

$\mathbf{p}^{(\lambda)}$ Translational momentum of body λ (or the spacecraft, if right superscript is missing).

$\mathbf{h}_i^{(\lambda)}$ Angular momentum of body λ (or the spacecraft, if right superscript is missing) about point O^i.

$\mathbf{f}^{(\lambda)}$ Force acting on body λ (or the spacecraft, if the right superscript is missing).

$u^{(\lambda)}$ Control torque acting on body λ about rotation axis $\hat{e}_i^1$.

$\mathbf{T}^{i,j}$ Transformation matrix from $\mathcal{F}^j$ to $\mathcal{F}^i$ coordinates.

$\mathbf{q}^{i,j}$ Quaternion describing the rotation of frame $\mathcal{F}^i$ into frame $\mathcal{F}^j$.

$\boldsymbol{\omega}^j$ Angular velocity vector of frame $\mathcal{F}^j$ about inertial frame $\mathcal{F}^1$.

$\boldsymbol{\Omega}^j$ Angular velocity vector of frame $\mathcal{F}^j$ about frame $\mathcal{F}^1$.

$\mathbf{E}$ Unit dyadic.

$\mathbf{E}_n$ $n \times n$ identity matrix.

$\mathbf{0}_{m,n}$ $n \times n$ zero matrix. If the subscript is missing, it indicates the three-dimensional null vector.

$\delta(\dots)$ Instantaneous knowledge error of a given variable, obtained as the subtraction of the measure/estimate from the true value.

$S[\dots]$ Power Spectral Density (PSD) of a given variable.

$(...)$ Mean value of a given variable, averaged over its total duration.

$A(...)$ Peak amplitude of a given variable with respect to the mean value over its total duration.

Appendix B

Simplified spacecraft dynamics components

The composition of the vectors that form the equation $\dot{\mathbf{z}}$, $\mathbf{g}$ and $\mathbf{u}$ is the following:

$$\dot{\mathbf{z}} \triangleq \begin{Bmatrix} {}^{1}\omega^{1} \\ \dot{\Omega}^{A} \end{Bmatrix}, \quad \mathbf{g} \triangleq \begin{Bmatrix} \mathbf{g}^{(B)} \\ \mathbf{g}^{(A)} \end{Bmatrix}, \quad \mathbf{u} \triangleq \begin{Bmatrix} \mathbf{u}^{(W)} \\ -\mathbf{u}^{(A)} \end{Bmatrix} \tag{B.1}$$

Components of $\mathbf{g}$ are as follows:

$$\begin{aligned} g^{(A)} \triangleq \ \hat{\mathbf{e}}_z^A \cdot \Big\{ & m_A(\boldsymbol{\omega}^1 \times \mathbf{r}_{A,}) \times (\boldsymbol{\omega}^1 \times \mathbf{r}_{1,A}) - \widetilde{\boldsymbol{\omega}}^1 \cdot (\mathbf{J}_{1/A}^{(A)} + \mathbf{I}_A^{(A)}) \cdot \boldsymbol{\omega}^1 \\ & - \widetilde{\boldsymbol{\omega}}^1 \cdot \mathbf{I}_A^{(A)} \cdot \Omega^A \hat{\mathbf{e}}_z^z \Big\} \end{aligned} \tag{B.2}$$

$$\begin{aligned} g^{(B)} \triangleq \ & - m\mathbf{r}_{1,} \times \Big[(\boldsymbol{\omega}^1 \times \mathbf{r}_{1,}) \times \boldsymbol{\omega}^1 \Big] - \widetilde{\boldsymbol{\omega}}^1 \cdot \mathbf{I}_1 \cdot \boldsymbol{\omega}^1 - \sum_{\lambda \in W} \widetilde{\boldsymbol{\omega}}^1 \cdot \mathbf{I}_{\lambda,\mathrm{rot}}^{(\lambda)} \Omega^\lambda \hat{\mathbf{e}}_z^\lambda \\ & - \widetilde{\boldsymbol{\omega}}^1 \cdot \left(\mathbf{J}_{1/A}^{(A)} + \mathbf{I}_A^{(A)} \right) \cdot \Omega^A \hat{\mathbf{e}}_z^A \end{aligned} \tag{B.3}$$

Components of $\mathbf{u}$ are as follows:

$$u^{(A)} \triangleq \hat{\mathbf{e}}_z^A \cdot \mathbf{u}^{(A)} \tag{B.4}$$

$$u^{(W)} \triangleq \sum_{\lambda \in W} u^{(\lambda)} \hat{\mathbf{e}}_z^\lambda \tag{B.5}$$

Mass matrix $\mathbf{M}$ has the following form:

$$\mathbf{M} \triangleq \begin{bmatrix} {}^{1}\mathbf{I}_1 + m^1 \mathbf{r}_{1,}^1 \mathbf{r}_1^T - \sum_{\lambda \in \mathcal{W}} \mathbf{I}_{\lambda,\mathrm{rot}}^{(A)} \, {}^{1^A}\mathbf{e}_z \left({}^{1^A}\mathbf{e}_z \right)^T & \left({}^{1^A}\mathbf{e}_z \right)^T \left({}^{J^{(A)}}_{1/A} + \mathbf{I}_A^{(A)} \right) \\ \left({}^{J^{(A)}}_{1/A} + \mathbf{I}_A^{(A)} \right) {}^{1^A}\mathbf{e}_z & \left({}^{1^A}\mathbf{e}_z \right)^T \mathbf{I}_A^{(A)} {}^{1^A}\mathbf{e}_z \end{bmatrix}$$

Appendix C

Complete algorithm

The complete filter algorithm is given in this appendix:

```
function [x_hat, P] = UKF(z, x_old, P_old, OM_lambda, u_lambda,
   u_f, ... % -> inputs
                          H, R, Mtot, MA, r_1A_1, SmodProp, ... %
    -> system parameters
                          ez_2_1, ez_3_1, ez_4_1, ez_5_1, ez_6_1,
    ez_A_1, W_UKF, tau_filt, Q) % -> rotation axes
% z: measurements
% u_f: external control input
% qI1 = x(1:4);
% thA = x(5);
% om1 = x(6:8);
% OMA = x(9);

n_UKF = length(x_old);
m_UKF = length(z);

f = @(x) ...
... kinematics
... qI1_dot =
[0.5*[-x(2) -x(3) -x(4);
x(1) -x(4) x(3);
x(4) x(1) -x(2);
-x(3) x(2) x(1)]*x(6:8);
... thetaA_dot =
x(9);
... dynamics
SmodProp.InvM*[...
... g1 =
-(Mtot*skewsym(SmodProp.r_1c_1)*skewsym(skewsym(x(6:8))*SmodProp.
   r_1c_1) + skewsym(x(6:8))*SmodProp.I_1_1)*x(6:8) + ...
- u_lambda(1)*ez_2_1 + skewsym(x(6:8))*SmodProp.I2_2_1*OM_lambda
   (1)*ez_2_1 + ...
- u_lambda(2)*ez_3_1 + skewsym(x(6:8))*SmodProp.I3_3_1*OM_lambda
   (2)*ez_3_1 + ...
- u_lambda(3)*ez_4_1 + skewsym(x(6:8))*SmodProp.I4_4_1*OM_lambda
   (3)*ez_4_1 + ...
```

```
- u_lambda(4)*ez_5_1 + skewsym(x(6:8))*SmodProp.I5_5_1*OM_lambda
   (4)*ez_5_1 + ...
- u_lambda(5)*ez_6_1 + skewsym(x(6:8))*SmodProp.I6_6_1*OM_lambda
   (5)*ez_6_1 + ...
- skewsym(x(6:8))*(SmodProp.JA_1A_1+SmodProp.IA_A_1)*x(9)*ez_A_1
   + ...
+ u_f(1:3);
... gA =
ez_A_1'*(MA*skewsym(skewsym(SmodProp.r_AcA_1)*x(6:8))*(skewsym(
   r_1A_1)*x(6:8)) ...
- skewsym(x(6:8))*(SmodProp.JA_A1_1+SmodProp.IA_A_1)*x(6:8) + ...
- skewsym(x(6:8))*SmodProp.IA_A_1*x(9)*ez_A_1) + ...
+ u_f(4)]];

%%% UKF %%%
% sigma points
X = zeros(n_UKF,2*n_UKF+1);
X(:,1) = x_old;

matX = real(sqrtm(n_UKF/(1-W_UKF(1))*P_old));

for i = 2:n_UKF+1
    X(:,i) = x_old + matX(:,i-1);
    X(1:4,i) = X(1:4,i)/norm(X(1:4,i));
    X(:,i+n_UKF) = x_old - matX(:,i-1);
    X(1:4,i+n_UKF) = X(1:4,i+n_UKF)/norm(X(1:4,i+n_UKF));
end

% state prediction
X_pred = zeros(n_UKF,2*n_UKF+1);
x_pred = zeros(n_UKF,1);
P_pred = zeros(n_UKF);
for i = 1:2*n_UKF+1
    X_pred(:,i) = X(:,i) + tau_filt*f(X(:,i));
    X_pred(1:4,i) = X_pred(1:4,i)/norm(X_pred(1:4,i));
    x_pred = x_pred + W_UKF(i)*X_pred(:,i);
end
x_pred(1:4) = x_pred(1:4)/norm(x_pred(1:4));
for i = 1:2*n_UKF+1
    P_pred = P_pred + W_UKF(i)*(X_pred(:,i)-x_pred)*(X_pred(:,i)-
   x_pred)';
end

P_pred = P_pred + Q;

if min(eig(P_pred)) <= 0
    error('P_pred non positiva');
end

% measurement prediction and cross-covariances
Z = zeros(m_UKF,2*n_UKF+1);
z_pred = zeros(m_UKF,1);
S_zz = zeros(m_UKF);
```

```
78 S_xz = zeros(n_UKF,m_UKF);
79 for i = 1:2*n_UKF+1
80     Z(:,i) = H*X_pred(:,i);
81     Z(1:4,i) = Z(1:4,i)/norm(Z(1:4,i));
82     z_pred = z_pred + W_UKF(i)*Z(:,i);
83 end
84 z_pred(1:4) = z_pred(1:4)/norm(z_pred(1:4));
85 for i = 1:2*n_UKF+1
86     S_zz = S_zz + W_UKF(i)*(Z(:,i)-z_pred)*(Z(:,i)-z_pred)';
87     S_xz = S_xz + W_UKF(i)*(X_pred(:,i)-x_pred)*(Z(:,i)-z_pred)';
88 end
89 S_zz = S_zz + R;
90
91 % update
92 K = S_xz/S_zz;
93 x_hat = x_pred + K*(z-z_pred);
94 P = P_pred - K*S_zz*K';
95
96 % quat norm
97 x_hat(1:4) = x_hat(1:4)/norm(x_hat(1:4));
98 end
```

Listing C.1: Complete UKF algorithm

Appendix D

Complete algorithm with Euler angle model

The complete filter algorithm with the Euler angle model is given in this appendix:

```
function [x_hat, P] = UKF(z, x_old, P_old, OM_lambda, u_lambda,
  u_f, ... % -> inputs
                          H, R, Mtot, MA, r_1A_1, SmodProp, ... %
   -> system parameters
                          ez_2_1, ez_3_1, ez_4_1, ez_5_1, ez_6_1,
   ez_A_1, W_UKF, tau_filt, Q) % -> rotation axes
% z: measurements
% u_f: external control input
% qI1 = x(1:4);
% thA = x(5);
% om1 = x(6:8);
% OMA = x(9);

n_UKF = length(x_old);
m_UKF = length(z);

f = @(x) ...
... kinematics
... eul =
                  [[0                 0                      -1
   ;...
                   sin(x(3)-x(1))   cos(x(3)-x(1))         0
   ;...
                   cos(x(3)-x(1))   -sin(x(3)-x(1))        0
   ]\x(5:7)*double(abs(x(2))==pi/2) +...
(1/cos(x(2)))*[cos(x(2)) sin(x(1))*sin(x(2)) cos(x(1))*sin(x(2))
   ;...
                   0      cos(x(1))*cos(x(2)) -sin(x(1))*cos(x(2))
   ;...
                   0           sin(x(1))             cos(x(2))
   ]*x(5:7)*double(abs(x(2))~=pi/2);
... thetaA_dot =
x(8);
... dynamics
SmodProp.InvM*[...
```

```
... g1 =
-(Mtot*skewsym(SmodProp.r_1c_1)*skewsym(skewsym(x(5:7))*SmodProp.
   r_1c_1) + skewsym(x(5:7))*SmodProp.I_1_1)*x(5:7) + ...
- u_lambda(1)*ez_2_1 + skewsym(x(5:7))*SmodProp.I2_2_1*OM_lambda
   (1)*ez_2_1 + ...
- u_lambda(2)*ez_3_1 + skewsym(x(5:7))*SmodProp.I3_3_1*OM_lambda
   (2)*ez_3_1 + ...
- u_lambda(3)*ez_4_1 + skewsym(x(5:7))*SmodProp.I4_4_1*OM_lambda
   (3)*ez_4_1 + ...
- u_lambda(4)*ez_5_1 + skewsym(x(5:7))*SmodProp.I5_5_1*OM_lambda
   (4)*ez_5_1 + ...
- u_lambda(5)*ez_6_1 + skewsym(x(5:7))*SmodProp.I6_6_1*OM_lambda
   (5)*ez_6_1 + ...
- skewsym(x(5:7))*(SmodProp.JA_1A_1+SmodProp.IA_A_1)*x(8)*ez_A_1
   + ...
+ u_f(1:3);
... gA =
ez_A_1'*(MA*skewsym(skewsym(SmodProp.r_AcA_1)*x(5:7))*(skewsym(
   r_1A_1)*x(5:7)) ...
- skewsym(x(5:7))*(SmodProp.JA_A1_1+SmodProp.IA_A_1)*x(5:7) + ...
- skewsym(x(5:7))*SmodProp.IA_A_1*x(8)*ez_A_1) + ...
+ u_f(4)]];

%%% UKF %%%
% sigma points
X = zeros(n_UKF,2*n_UKF+1);
X(:,1) = x_old;

matX = real(sqrtm(n_UKF/(1-W_UKF(1))*P_old));

for i = 2:n_UKF+1
    X(:,i) = x_old + matX(:,i-1);
    X(:,i+n_UKF) = x_old - matX(:,i-1);
end

% state prediction
X_pred = zeros(n_UKF,2*n_UKF+1);
x_pred = zeros(n_UKF,1);
P_pred = zeros(n_UKF);
for i = 1:2*n_UKF+1
    X_pred(:,i) = X(:,i) + tau_filt*f(X(:,i));
    x_pred = x_pred + W_UKF(i)*X_pred(:,i);
end

for i = 1:2*n_UKF+1
    P_pred = P_pred + W_UKF(i)*(X_pred(:,i)-x_pred)*(X_pred(:,i)-
   x_pred)';
end

P_pred = P_pred + Q;

if min(eig(P_pred)) <= 0
    error('P_pred non positiva');
```

```
71 end
72
73 % measurement prediction and cross-covariances
74 Z = zeros(m_UKF,2*n_UKF+1);
75 z_pred = zeros(m_UKF,1);
76 S_zz = zeros(m_UKF);
77 S_xz = zeros(n_UKF,m_UKF);
78 for i = 1:2*n_UKF+1
79     Z(:,i) = H*X_pred(:,i);
80     z_pred = z_pred + W_UKF(i)*Z(:,i);
81 end
82
83 for i = 1:2*n_UKF+1
84     S_zz = S_zz + W_UKF(i)*(Z(:,i)-z_pred)*(Z(:,i)-z_pred)';
85     S_xz = S_xz + W_UKF(i)*(X_pred(:,i)-x_pred)*(Z(:,i)-z_pred)';
86 end
87 S_zz = S_zz + R;
88
89 % update
90 K = S_xz/S_zz;
91 x_hat = x_pred + K*(z-z_pred);
92 P = P_pred - K*S_zz*K';
93
94 end
```

Listing D.1: Complete UKF algorithm with Eulero angle model

Appendix E

Simulation parameters

Antenna Assembly A

- Mass $m_A = 68.831\,\mathrm{kg}$
- Inertia $\mathbf{I}_A^{(A)} = \begin{bmatrix} 171.664 & 0 & 0 \\ 0 & 171.664 & 0 \\ 0 & 0 & 64.115 \end{bmatrix} \mathrm{kg \cdot m^2}$
- Rotation axis $\hat{\mathbf{e}}_z^A = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T$
- Position of point O_A: $\mathbf{r}_{1,A} = \begin{bmatrix} 0 \\ 0 \\ -1.375 \end{bmatrix}^T \mathrm{m}$
- Center of mass position: $\mathbf{r}_{A,A} = \begin{bmatrix} 0 \\ 0 \\ -1.25 \end{bmatrix}^T \mathrm{m}$

Balancing Mass 7

- Mass $m_7 = 11.719\,\mathrm{kg}$
- Inertia $\mathbf{I}_A^{(7)} = \begin{bmatrix} 1.294 \times 10^{-1} & 0 & -1.0051 \\ 0 & 9.672 & 0 \\ -1.0051 & 0 & 9.566 \end{bmatrix} \mathrm{kg \cdot m^2}$
- Center of mass position: $\mathbf{r}_{A,7} = \begin{bmatrix} -9.023 \times 10^{-1} \\ 0 \\ -9.505 \times 10^{-2} \end{bmatrix}^T \mathrm{m}$

Antenna Reflector 8

- Mass $m_8 = 57.112\,\mathrm{kg}$
- Inertia $\mathbf{I}_A^{(8)} = \begin{bmatrix} 171.535 & 0 & 1.0051 \\ 0 & 161.992 & 0 \\ 1.0051 & 0 & 54.549 \end{bmatrix} \mathrm{kg \cdot m^2}$

- Position of point O_8: $\mathbf{r}_{A,8} = \begin{bmatrix} 0 \\ 0 \\ -1.25 \end{bmatrix}^T$ m
- Center of mass position: $\mathbf{r}_{8,8} = \begin{bmatrix} 1.852 \times 10^{-1} \\ 0 \\ -2.37 \times 10^{-1} \end{bmatrix}^T$ m

Body 1

- Mass $m_1 = 996.2\,\text{kg}$
- Inertia $\mathbf{I}_1^{(1)} = \begin{bmatrix} 1175 & 0 & 0 \\ 0 & 1528 & 0 \\ 0 & 0 & 893.2 \end{bmatrix}$ kg · m^2

Wheels $\lambda \in \{2, 3, 4, 5\}$ (Honeywell HR14-50)

- Mass $m_\lambda = 8.5\,\text{kg}$
- Rotation axes:
  - $\hat{\mathbf{e}}_{1,2} = \begin{bmatrix} 0.5 \\ 0.5 \\ \frac{\sqrt{2}}{2} \end{bmatrix}^T$
  - $\hat{\mathbf{e}}_{1,3} = \begin{bmatrix} -0.5 \\ 0.5 \\ \frac{\sqrt{2}}{2} \end{bmatrix}^T$
  - $\hat{\mathbf{e}}_{1,4} = \begin{bmatrix} -0.5 \\ -0.5 \\ \frac{\sqrt{2}}{2} \end{bmatrix}^T$
  - $\hat{\mathbf{e}}_{1,5} = \begin{bmatrix} 0.5 \\ -0.5 \\ \frac{\sqrt{2}}{2} \end{bmatrix}^T$
- Positions of points O_λ:
  - $\mathbf{r}_{1,2} = \begin{bmatrix} 0.55 \\ 0.325 \\ 0 \end{bmatrix}^T$ m
  - $\mathbf{r}_{1,3} = \begin{bmatrix} -0.55 \\ 0.325 \\ 0 \end{bmatrix}^T$ m
  - $\mathbf{r}_{1,4} = \begin{bmatrix} -0.55 \\ -0.325 \\ 0 \end{bmatrix}^T$ m
  - $\mathbf{r}_{1,5} = \begin{bmatrix} 0.55 \\ -0.325 \\ 0 \end{bmatrix}^T$ m

- Center of mass position: $\mathbf{r}_{\lambda,\lambda} = \begin{bmatrix} 4.121 \times 10^{-1} \\ 0 \\ -9.946 \times 10^{-6} \end{bmatrix}^T \mu\text{m}$
- Inertia $\mathbf{I}_\lambda^{(\lambda)} = \begin{bmatrix} 4.176 \times 10^{-2} & 0 & 9.1 \times 10^{-7} \\ 0 & 4.176 \times 10^{-2} & 0 \\ 9.1 \times 10^{-7} & 0 & 7.947 \times 10^{-2} \end{bmatrix} \text{kg} \cdot \text{m}^2$
- Maximum speed: $\pm 6000\,\text{rpm}$
- Maximum torque: $0.2\,\text{N} \cdot \text{m}$

Wheel 6 (Honeywell HR16-100)

- Mass $m_6 = 12\,\text{kg}$
- Inertia $\mathbf{I}_6^{(6)} = \begin{bmatrix} 8.243 \times 10^{-2} & 0 & 1.54 \times 10^{-6} \\ 0 & 8.243 \times 10^{-2} & 0 \\ 1.54 \times 10^{-6} & 0 & 1.592 \times 10^{-1} \end{bmatrix} \text{kg} \cdot \text{m}^2$
- Rotation axis $\hat{\mathbf{e}}_{1,6} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T$
- Position of point O_6: $\mathbf{r}_{1,6} = \begin{bmatrix} 0 \\ 0 \\ -0.6875 \end{bmatrix}^T \text{m}$
- Center of mass position: $\mathbf{r}_{6,6} = \begin{bmatrix} 4 \times 10^{-1} \\ 0 \\ -8.028 \times 10^{-6} \end{bmatrix}^T \mu\text{m}$
- Maximum speed: $\pm 6000\,\text{rpm}$
- Maximum torque: $0.2\,\text{N} \cdot \text{m}$

Simulation Parameters

- Simulation duration: $11400\,\text{s}$
- Integration step: $5 \times 10^{-4}\,\text{s}$

Initial Conditions

- Quaternion from F_I to F_1: $\text{ıı}_{I,1} = \begin{bmatrix} 0.8642 \\ -5.6 \times 10^{-2} \\ -3.23 \times 10^{-2} \\ 0.5 \end{bmatrix}^T$
- Quaternions from F_1 to the wheel frames:

$$- \text{ıı}_{1,2} = 10^{-1} \cdot \begin{bmatrix} 8.536 \\ -1.464 \\ 3.536 \\ 3.536 \end{bmatrix}^T$$

$$- \text{ıı}_{1,3} = 10^{-1} \cdot \begin{bmatrix} 3.536 \\ -3.536 \\ 1.464 \\ 8.536 \end{bmatrix}^T$$

$$- \text{ıı}_{1,4} = 10^{-1} \cdot \begin{bmatrix} 3.536 \\ 3.536 \\ 1.464 \\ -8.536 \end{bmatrix}^T$$

$$- \text{ıı}_{1,5} = 10^{-1} \cdot \begin{bmatrix} 8.536 \\ 1.464 \\ 3.536 \\ -3.536 \end{bmatrix}^T$$

- Quaternion from F_1 to F_6: $\text{ıı}_{1,6} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T$
- Quaternion from F_1 to F_A: $\text{ıı}_{1,A} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T$
- Absolute angular velocity of F_1: $\boldsymbol{\omega}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ rad/s
- Absolute angular velocity of the wheels $\lambda \in \mathcal{W}$:

$$\omega_\lambda = 0\,\text{rad/s} \quad \text{about axis } \hat{\mathbf{e}}_\lambda$$

- Absolute angular velocity of antenna A:

$$\omega_A = \frac{2\pi}{5}\,\text{rad/s} \quad \text{about axis } \hat{\mathbf{e}}_A$$

- Absolute position of the satellite's center of mass (S/C):

$$\mathbf{r}_{I,1} = \begin{bmatrix} 0 \\ 0 \\ 6.878 \times 10^6 \end{bmatrix}^T \text{m}$$

- Absolute velocity of the satellite (S/C):

$$\mathbf{v}_{I,1} = \begin{bmatrix} 7.613 \times 10^3 \\ 0 \\ 0 \end{bmatrix}^T \text{m/s}$$

Controller Parameters

- Control frequency: 5 Hz
- Gain matrix $\mathbf{K}_1 = \begin{bmatrix} 0.6\mathbf{E} & 0 \\ 0 & 5 \end{bmatrix}$
- Gain matrix $\mathbf{K}_2 = \begin{bmatrix} 0.1\mathbf{E} & 0 \\ 0 & 3 \end{bmatrix}$
- Slope of the tanh function: $\eta = 1$

Bibliography

[1] ESA. *Cairt and Wivern Earth Explorer candidates go forward*. 2023. URL: https://www.esa.int/Applications/Observing_the_Earth/FutureEO/Cairt_and_Wivern_Earth_Explorer_candidates_go_forward (cit. on p. 1).

[2] Polito. *WIVERN*. 2023. URL: https://wivern.polito.it/ (visited on 04/02/2025) (cit. on p. 1).

[3] European Space Agency. *Report for Mission Selection: Earth Explorer 11 Candidate Mission WIVERN*. Tech. rep. ESA-EOPSM-WIVE-RP-4798. Issue 1.0, 06 June 2025. Noordwijk, The Netherlands: European Space Agency, 2025. DOI: 10.5281/zenodo.15607041 (cit. on pp. 1, 2).

[4] Frederic Tridon, Alessandro Battaglia, Ali Rizik, FE Scarsi, and A Illingworth. «Filling the gap of wind observations inside tropical cyclones». In: *Earth and Space Science* 10.11 (2023), e2023EA003099 (cit. on p. 1).

[5] AJ Illingworth et al. «WIVERN: A new satellite concept to provide global in-cloud winds, precipitation, and cloud properties». In: *Bulletin of the American Meteorological Society* 99.8 (2018), pp. 1669–1687 (cit. on pp. 1–3).

[6] Alessandro Battaglia et al. «Peeping inside tropical cyclones with the WIVERN-2 space-borne Doppler radar». In: *ESA Living Planet Symposium 2022*. Presented at the ESA Living Planet Symposium 2022. European Space Agency (ESA). Bonn, Germany, 2022. URL: https://wivern.polito.it/wp-content/uploads/2025/06/TC_WIVERN_observations-3.pdf (cit. on p. 1).

[7] Alessandro Battaglia, Paolo Martire, Eric Caubet, Laurent Phalippou, Fabrizio Stesina, Pavlos Kollias, and Anthony Illingworth. «Observation error analysis for the WInd VElocity Radar Nephoscope W-band Doppler conically scanning spaceborne radar via end-to-end simulations». In: *Atmospheric Measurement Techniques* 15.9 (2022), pp. 3011–3030 (cit. on pp. 1, 3).

[8] Richard J Doviak, Dusan S Zrnic, and Dale S Sirmans. «Doppler weather radar». In: *Proceedings of the IEEE* 67.11 (1979), pp. 1522–1553 (cit. on p. 3).

[9] Alessandro Battaglia, Simone Tanelli, and Pavlos Kollias. «Polarization diversity for millimeter spaceborne Doppler radars: An answer for observing deep convection?» In: *Journal of Atmospheric and Oceanic Technology* 30.12 (2013), pp. 2768–2787 (cit. on p. 3).

[10] Filippo Emilio Scarsi, Alessandro Battaglia, Frederic Tridon, Paolo Martire, Ranvir Dhillon, and Anthony Illingworth. «Mispointing characterization and Doppler velocity correction for the conically scanning WIVERN Doppler radar». In: *Atmospheric Measurement Techniques* 17.2 (2024), pp. 499–514 (cit. on pp. 3, 4).

[11] Francesco Manconi, Paolo Martire, Fabrizio Stesina, and Alessandro Battaglia. «High accuracy attitude determination of a spacecraft with a fast-rotating Doppler radar reflector». In: *Acta Astronautica* 233 (2025), pp. 66–81 (cit. on pp. 3, 4, 9, 10, 12–14, 25–27, 29, 35, 36, 41, 54, 63, 72, 73).

[12] Stanisław Konatowski, Piotr Kaniewski, and Jan Matuszewski. «Comparison of estimation accuracy of EKF, UKF and PF filters». In: *Annual of Navigation* 23 (2016), pp. 69–87 (cit. on pp. 4, 36, 43).

[13] John L Crassidis and F Landis Markley. «Unscented filtering for spacecraft attitude estimation». In: *Journal of guidance, control, and dynamics* 26.4 (2003), pp. 536–542 (cit. on pp. 4, 62).

[14] Shaoke Yang and Hongxin Li. «Application of EKF and UKF in target tracking problem». In: *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. Vol. 1. IEEE. 2016, pp. 116–120 (cit. on pp. 4, 26).

[15] Pavithra Kasula, James F Whidborne, and Zeeshan A Rana. «Quaternion-based attitude estimation of an aircraft model using computer vision». In: *Sensors* 24.12 (2024), p. 3795 (cit. on p. 12).

[16] Claude Sammut and Geoffrey I Webb. *Encyclopedia of machine learning*. Springer Science & Business Media, 2011 (cit. on p. 27).

[17] L. J. Romans. *Optimal Combination of Quaternions from Multiple Star Cameras*. Tech. rep. JPL Interoffice Memorandum, 2003. URL: `https://archive.podaac.earthdata.nasa.gov/965%20podaac-ops-cumulus-docs/grace/open/docs/quaternion_memo.pdf` (cit. on p. 29).

[18] Maria Isabel Ribeiro. «Kalman and extended kalman filters: Concept, derivation and properties». In: *Institute for Systems and Robotics* 43.46 (2004), pp. 3736–3741 (cit. on pp. 34, 35).

[19] Greg Welch, Gary Bishop, et al. «An introduction to the Kalman filter». In: (1995) (cit. on p. 35).

[20] Keisuke Fujii. «Extended kalman filter». In: *Refernce Manual* 14 (2013), p. 41 (cit. on p. 35).

[21] Paul Zarchan and Howard Musoff. *Fundamentals of Kalman Filtering: A Practical Approach*. Vol. 232. American Institute of Aeronautics and Astronautics, 2009 (cit. on p. 35).

[22] Eric A Wan and Rudolph Van Der Merwe. «The unscented Kalman filter for nonlinear estimation». In: *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*. Ieee. 2000, pp. 153–158 (cit. on p. 42).

[23] Kai Xiong, HY Zhang, and CW Chan. «Performance evaluation of UKF-based nonlinear filtering». In: *Automatica* 42.2 (2006), pp. 261–270 (cit. on p. 42).

[24] Simon J Julier and Jeffrey K Uhlmann. «New extension of the Kalman filter to nonlinear systems». In: *Signal processing, sensor fusion, and target recognition VI*. Vol. 3068. Spie. 1997, pp. 182–193 (cit. on pp. 42–45).

[25] A Terejanu Gabriel. «Unscented kalman filter tutorial». In: *University at Buffalo, Buffalo* 42 (2011) (cit. on pp. 42, 43, 45, 62).

[26] James Fisher and SR Vadali. «Gyroless attitude control of multibody satellites using an unscented Kalman filter». In: *Journal of guidance, control, and dynamics* 31.1 (2008), pp. 245–251 (cit. on p. 45).

[27] Rudolph Van Der Merwe and Eric A Wan. «The square-root unscented Kalman filter for state and parameter-estimation». In: *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*. Vol. 6. IEEE. 2001, pp. 3461–3464 (cit. on p. 62).

[28] Fuming Sun, Guanglin Li, and Jingli Wang. «Unscented kalman filter using augmented state in the presence of additive noise». In: *2009 IITA International Conference on Control, Automation and Systems Engineering (case 2009)*. IEEE. 2009, pp. 379–382 (cit. on p. 77).