POLITECNICO DI TORINO

MASTER's Degree in MECHATRONIC ENGINEERING

MASTER's Degree Thesis

VISION-BASED POSE ESTIMATION OF AN UNKNOWN AND UNCOOPERATIVE OBJECT FOR ON-ORBIT SERVICING AND DEBRIS REMOVAL APPLICATIONS

Supervisors

Prof. MARTINA MAMMARELLA

CESARE DONATI

Prof. FABRIZIO DABBENE

Candidate

MICHELE COCO

OCTOBER 2024

Summary

The thesis is centered around the application of a sensor fusion algorithm for satellite tracking. The specific application considered is the PRISMA mission, which involves a complex task of autonomous satellite docking. In this mission, a MAIN satellite is responsible for tracking and eventually docking with a TARGET satellite. To model and simulate this docking scenario, an orbital simulator was developed in MATLAB. The choice of the sensor fusion algorithm comes from a broader understanding of the principal estimation techniques. The search starts from the classical Kalman Filter (KF). In particular the main motivation behind this work stems from two key challenges that arise in pose tracking; namely the need to model nonlinear dynamics that cannot be effectively treated by traditional filtering methods like the KF and non-Gaussian sensor noise. This noise deviates from the standard Gaussian assumptions that traditional filters rely on, leading to suboptimal performance in estimating the satellite's pose. In an effort to address these challenges, several filtering algorithms were explored and ultimately, the Unscented Kalman Filter (UKF) was selected as the foundation for this research due to its ability to accurately handle nonlinear systems without the need for linearization. First, the UKF was integrated with a smoother to deal with high-frequency noise in the system and provide a cleaner trajectory. One of the key contributions of this thesis is the incorporation of Huber's function into the filtering process, which provides robustness against outliers in the sensor measurements. In many real-world tracking problems, sensor data can contain significant outliers due to temporary failures or unexpected anomalies in the environment. By introducing Huber's function, the filter assigns adaptive weights to the measurements based on their deviation from the predicted value. Measurements that are close to the expected value receive higher weights, while those that deviate significantly (potential outliers) are assigned lower weights. This weighting system is controlled by a threshold parameter, which helps to balance the trade-off between being sensitive to actual changes in the state and ignoring spurious outliers. To test and validate the proposed algorithm, extensive simulations were carried out in two different scenarios. The first involved a relatively simple case using a double integrator model, which allowed for an initial evaluation of the filter's performance.

Next, the algorithm was applied to the full satellite dynamics, including both the orbital and attitude models. In these simulations, non-Gaussian noise was introduced in the measurements using a Gaussian mixture model. This model consists of two distinct Gaussian distributions with zero mean but very different variances. Each measurement was drawn from one of the two distributions with a certain probability, reflecting the variability and unpredictability of the sensor noise in real-world applications. The simulations confirmed the effectiveness of the Huber-based UKF (HUKF) in addressing both the nonlinear dynamics of the satellite and the non-Gaussian noise in the sensor measurements and of the Huber-based UKS (HUKS) to smooth the trajectory. Overall, the proposed method offers a robust and efficient solution for satellite pose tracking, particularly in scenarios involving complex dynamics and unreliable sensor data.

Acknowledgements

I would like to express my gratitude to professors Martina Mammarella and Fabrizio Dabbene, Phd student Cesare Donati and the AIKO team for their support throughout the development of this project. Your constant availability was crucial in helping me navigate through all challenges. My heartfelt thanks go to my family, especially my grandmothers, whose phone calls brightened my evenings. I can not leave out all my friends both close and faraway. In particular, the students from the campus Codegone, who put up with me for three years, and my schoolmates from Rome.

Table of Contents

List of Tables

List of Figures

Acronyms

KF
Kalman Filter

EKF
Extended Kalman Filter

DDF
Divided Difference Filter

CDF
Central Difference Filter

UKF
Unscented Kalman Filter

PF
Particle Filter

FFBS
Feedforward Filter Backward Smoother

UKS
Unscented Kalman Filter

MSE
Mean Square Error

RMSE
Root Mean Square Error

HUKF

Huber-based Unscented Kalman Filter

HUKS

Huber-based Unscented Kalman Smoother

SSO

Sun Synchronous Orbit

ISL

Intersatellite Link

SSC

Swedish Space Corporation

GPS

Global Navigation System

IGRF

International Geomagnetic Reference Field

LTAN

Local Time of the Ascending Node

GNS

Guidance, Navigation and Control

ECI

Earth Centered Inertial

ECEF

Earth Centered Earth Fixed

LVLH

Local Vertical Local Horizontal

Chapter 1

Introduction

1.1 Pose estimation in spatial applications

With the advancements in space exploration, the technology boundaries have been continuously pushed further away. This allows for increasingly complex operations to be conducted autonomously and efficiently in outer space such us repairing spacecrafts in orbit, debris removal and supplying orbital platforms and stations. To accomplish these tasks, complicated maneuvers need to be performed. For example, the phasing phase in the rendezvous and docking process, i.e., the stage during which the chaser spacecraft adjusts its orbit to catch up with the target spacecraft, as shown in Figure 1.1.

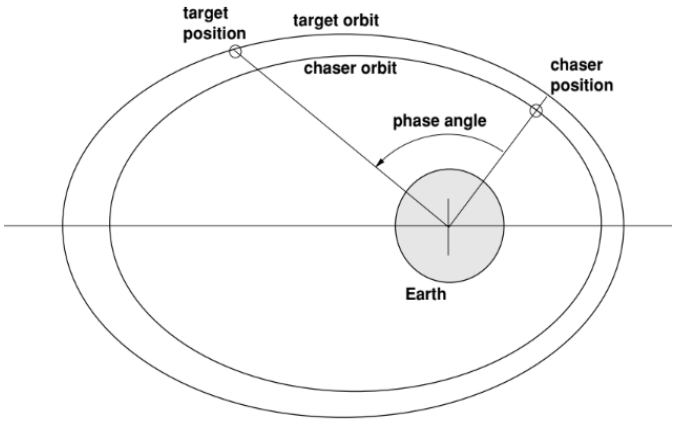

Figure 1.1: Visual representation of the phasing stage in the rendezvous process (credit: [1]).

This phase is critical because it guarantees that the two spacecraft will meet at

a certain location along the same orbit. To do so, the chaser must travel along a narrower orbit than the target. Thus, the phase angle, which is the angular separation between the two spacecrafts in their orbits, is gradually reduced. During phasing, the absolute position coordinates (described in the Earth Centered Inertial frame) of the chaser must be monitored to ensure it lies on the orbit that allows it to reach the target [1]. On the other hand, during the final stages of the mating process, as shown in Figure 1.2, between the two satellites an accurate measurement of the orientation is needed. Both spacecraft have specific docking ports that must align with each other to ensure a successful connection and even a slight misalignment in orientation could prevent the docking mechanisms from engaging correctly. Proper orientation guarantees that the docking interfaces are positioned correctly for a secure lock.

Figure 1.2: Chaser spacecraft approaching the target (credit: [2]).

In conclusion, pose estimation of a satellite is not only crucial for tracking its position and orientation along its orbital path, but also plays a fundamental role in enabling the successful execution of complex procedures. It ensures that the satellite can navigate accurately, approach other spacecraft safely, and achieve the delicate mechanical connections necessary for mission-critical tasks.

1.2 The role of estimators

Estimators play a crucial role in filtering noisy sensor data and predicting the state of the satellite over time. Various types of estimators can be used, each based on different mathematical principles. **Least Squares** (LS) [3] is one of the most basic and widely used estimation techniques, particularly for solving linear regression problems. It finds the parameter values that minimize the sum of squared differences between observed data and model predictions. The standard LS approach assumes that the noise or error has a constant variance and normally distributed. LS is the default method for fitting a straight line (or more complex linear models) to a set of data points. Mathematically, the goal is to minimize the residuals between the output y and the predicted output Hx, as described in equation 1.1

$$\hat{x} = \arg\min_x \|y - Hx\|^2, \tag{1.1}$$

where H is the measurement matrix function and x the state variable to estimate. LS assumes the errors are distributed normally, but if this assumption does not hold (for instance, if errors have varying variances), LS can provide biased estimates, leading to the need for more sophisticated techniques like **Weighted Least Squares** (WLS) [4]. It extends LS by assigning different weights to each observation, reflecting the confidence or precision of each measurement. WLS is especially useful when the errors in the data have different variances. The WLS estimator minimizes the weighted sum of squared residuals, allowing for more accurate parameter estimation when the error variances are not constant

$$\hat{x} = \arg\min_x (y - Hx)^T W (y - Hx), \tag{1.2}$$

where W is a diagonal matrix where each element represents the inverse of the variance of the corresponding observation. This approach ensures that observations with lower noise (i.e., higher precision) contribute more to the final estimate. **Maximum Likelihood Estimation** (MLE) [5] is a powerful general-purpose method for estimating parameters of a probabilistic model. MLE maximizes the likelihood function, which represents the probability of the observed data given a set of parameters. Unlike LS, MLE is not restricted to linear models and can handle a wide variety of data distributions, making it more flexible. If x is the variable to estimate and $p(y|x)$ is the likelihood to observe the data given the variable, MLE seeks to maximize it

$$\hat{x} = \arg\max_x p(y \mid x), \tag{1.3}$$

while MLE is simple and intuitive, it doesn't account for prior knowledge or uncertainty about the parameters, making it less effective in real-time scenarios

where noise and dynamics affect the measurements. Recursive Bayesian estimation addresses the shortcomings of LS, WLS, and MLE for dynamic systems. It combines prior information with new measurements in an optimal and recursive manner, making it perfect for online applications like satellite pose tracking. In this framework, the goal is to update the posterior distribution of the satellite's state as new sensor data becomes available

$$p(x_t \mid y_1, \ldots, y_t) = \frac{p(y_t)p(y_t \mid x_t)p(x_t \mid y_1, \ldots, y_{t-1})}{p(y_t)}, \tag{1.4}$$

where, to find the posterior $p(x_t \mid y_1, \ldots, y_t)$, the likelihood $p(y_t \mid x_t)$ of the new observation given the current state, the probability of the current state $p(x_t \mid y_1, \ldots, y_{t-1})$ and measurement $p(y_t)$ to occur are needed. This recursive approach is foundational to Kalman filters, which are widely used in satellite systems for tracking the satellite's pose (position and orientation) over time. The Kalman Filter builds on the recursive Bayesian framework and is optimal for linear systems with Gaussian noise. It iteratively predicts the next state and updates it with incoming measurements. The system model is expressed as

$$x_{t+1} = Fx_t + Bu_t + w_t \tag{1.5}$$

$$y_t = Hx_t + v_t, \tag{1.6}$$

where F is the state transition matrix, u the control inputs, w and v are the modeled process and measurement noise. The Kalman Filter proceeds in two stages. First it predicts the state at the next time step based on the current state

$$\hat{x}^-_{t+1} = F\hat{x}_t + Bu_t \tag{1.7}$$

$$P^-_{t+1} = FP_tF^T + Q, \tag{1.8}$$

where P_t is the error covariance matrix, and Q represents process noise covariance. The next step is to incorporate the new measurements to correct the predicted state

$$K_{t+1} = P^-_{t+1}H^T(HP^-_{t+1}H^T + R)^{-1} \tag{1.9}$$

$$\hat{x}_{t+1} = \hat{x}^-_{t+1} + K_{t+1}(y_{t+1} - H\hat{x}^-_{t+1}), \tag{1.10}$$

where K is the Kalman gain, and R is the measurement noise covariance. This recursive approach allows for real-time pose tracking, making Kalman filters ideal for satellite navigation systems.

1.3 Robust nonlinear filtering

The estimation of a satellite pose has evident obstacles to overcome.

1.3.1 Sensor noise

The space environment introduces factors such as vacuum conditions, extreme temperature variations, and high levels of radiation that can affect sensor behavior in ways that are difficult to fully simulate during terrestrial testing. For example, as the name suggests, **magnetometers** [6] measure the magnetic field and find its vector direction and strength. In space, they rely on the calculation of the Earth's magnetic field to determine the satellite orientation. In particular they are most effective in low-Earth orbits (LEO), where the magnetic field is strong enough for accurate measurements. By comparing the detected magnetic field to known models, such as the International Geomagnetic Reference Field (IGRF), magnetometers allow satellites to determine their orientation relative to Earth's magnetic field. Thus, they are particularly important for low-altitude satellites, such as Earth observation satellites, where accurate attitude determination is needed to orient cameras, sensors, or communication antennas. In addition, they are useful for scientific missions studying the Earth's magnetic field, such as geomagnetic mapping missions or for detecting changes in magnetic activity, which can also help monitor space weather and the final moments in docking missions where the ports of both the target and the chaser satellite must be properly aligned. However, these devices are triggered by any magnetic source, including that generated by the spacecraft electrical system. Another important device that constitutes the navigation control system of a satellite is the **Sun sensor** [7]. They are key instruments for determining a satellite's orientation relative to the Sun. Their main function is to measure the angle of the incoming sunlight, which helps in attitude determination. One of the most critical applications of sun sensors is in safe mode operations, where the satellite relies on them to orient itself in the event of system failure. This ensures that the solar panels are aligned with the Sun, maintaining the power supply even when other systems malfunction. Additionally, sun sensors are widely used in solar-powered satellites to keep their solar arrays optimally aligned for maximum energy generation. They are also used in science missions where specific instruments need to be oriented toward the Sun, like in solar observation satellites or those studying solar radiation. Their simple design and low power consumption make sun sensors reliable for long-duration missions, particularly in deep space where sunlight is often the only available reference. This comes at the expenses of tolerance to perturbations. In particular, it is sensible to light reflection of the Sun from Earth, creating the so-called Albedo effect. This phenomenon can be modelled, but it depends on the position of the

satellite and the configuration between the celestial bodies. Lastly, the Global Positioning System **GPS** [8] receivers mounted on satellites are used to evaluate the absolute position. The GPS receiver onboard the satellite uses signals from at least four GPS satellites to determine its location through triangulation. Each GPS satellite sends a signal containing its position and the exact time the signal was transmitted. The receiver calculates its distance from each GPS satellite based on the time delay of these signals and, using this data, determines its precise position in an Earth-centered coordinate system, such as Earth-Centered Inertial (ECI) or Earth-Centered, Earth-Fixed (ECEF). By having onboard positional awareness, the satellite can autonomously adjust its orbit or perform maneuvers without needing constant control from ground stations. This is particularly useful in low Earth orbit missions or in constellations. For missions that involve multiple satellites flying in close formation, such as scientific observation missions, the GPS receiver helps maintain precise relative positions between the satellites, enabling coordinated operations. GPS receivers help satellites relay accurate positional data to ground stations, facilitating mission control and ensuring that the satellite can fulfill its mission objectives, such as imaging specific areas or communicating with ground antennas. The receiver's calculations are continuous, allowing for real-time position tracking. However they can experience thermal and radiation noise from their electronics, leading to small, random errors in the signal measurement. The three types of sensors are presented in Figure 1.3.

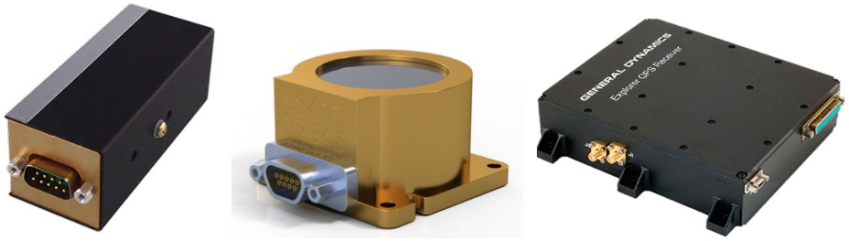

Figure 1.3: From left to right: MAG-3 magnetometer, AQUILA-D02 Sun Sensor, Explorer GPS Spaceborne Receiver (credit: [9], [10], [11]).

In space, thermal fluctuations and radiation can also affect the sensor performance. These disturbances can not be modeled to extreme precision causing a behaviour which is not Gaussian anymore. In fact, a Gaussian distribution is suitable to model data dispersion as long as all variable relations are known. When this assumption is not valid anymore, it is necessary to introduce robust algorithms that can handle outliers.

1.3.2 Nonlinear functions

Another problem to investigate is related to the system nonlinearity. A function is a map between two or more variables. In particular, probabilistic variables are represented by a distribution that spans over a certain range in the feature space. The input variable distribution is also called prior, while the output one is referred to as posterior. A nonlinear function (e.g., a trigonometric or exponential function) causes the relationship between the input and the output to be more complex. As a result, the output (posterior) may deviate significantly from the original type of distribution (prior). For example, if the prior distribution is Gaussian, a nonlinear function might transform it into a non-Gaussian posterior, making it harder to model using simple linear assumptions.
Traditional linear filters, such as the standard Kalman Filter, are inadequate, as they rely on linear assumptions that can lead to significant estimation errors [12]. Thus, a filter that can model as precise as possible the changes in the characteristics of a probability distribution undergoing a nonlinear transformation, is optimal in the pose estimation problem.

1.4 A panoramic view on nonlinear filters

The last section poses two problems in the use of the classical Kalman filter. Namely, it can not model a non linear function and it follows the assumption of Gaussian measurement noise. The first task is tackled by the nonlinear recursive Bayesian filters. They can be arranged by their method of reconstructing the posterior distribution as expressed in Figure 1.4. One way of measuring a filter performance is to analyze how well it estimates the moments of the distribution such as the mean and the variance. A common approach to compute the moments is to approximate the nonlinear system and measurement models.
The **Extended Kalman Filter** (EKF) [13] exploits first-order Taylor expansions around the prior state mean for model linearization. Another option is to use polynomial interpolations. This allows not to compute explicit derivatives. Such filters are for example the **Central Difference Filter** (CDF) [14] and the **Divided Difference Filter** (DDF) [15]. The problem with these approaches is that they fail to capture the second moment of the distribution, namely the variance [12]. Introducing a second order approximation term improves the performance but increases the computational complexity. Linear regression Kalman filters are sample-based approach. This means that the state densities are represented by a set of samples. The **Unscented Kalman Filter** (UKF) [16] employs $2n + 1$ (where n is the system dimension) samples to capture the prior distributions. Thus, the number of samples is fixed and it depends linearly on the system dimension number. A filter that does not rely on Kalman theory is the **Particle Filter** (PF)

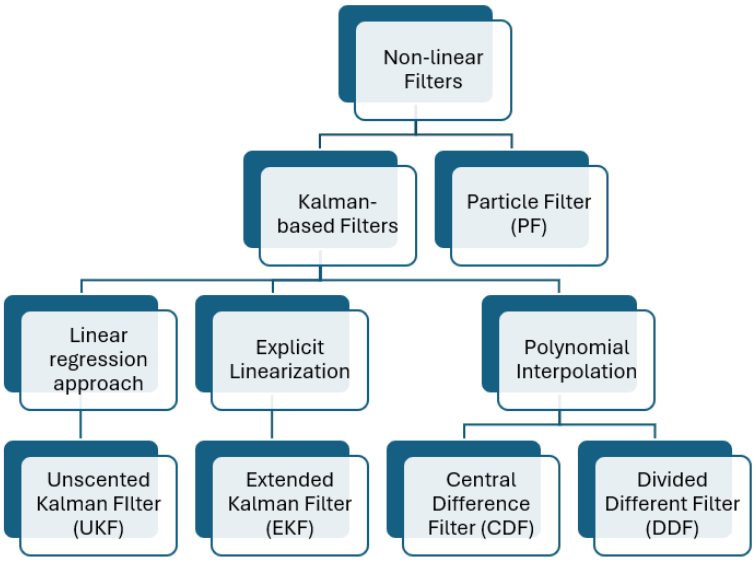

Figure 1.4: Non-linear filter classification by approaches.

[17]. This is already an advantage because it does not rely on any assumption on the noise distribution unlike the classical Kalman filter. It constructs the posterior distribution of the system's state as a set of weighted particles. Each particle represents a possible state of the system and is characterized by its state vector and an associated weight that indicates how likely that particle represents the true state given the observations. Particle Filters can represent complex, multimodal distributions and global localization problems (where the starting point is unknown). This is perfect for mapping a room in an autonomous mobile robot scenario. However, for satellites, the situation is quite different. The initial position is typically known (or at least roughly estimated) through GPS coordinates or similar telemetry systems. While GPS signals may contain noise, they still provide a reliable reference. A visual representation of the difference between a global localization problem and a tracking one is presented in Figure 1.5. This transforms the problem into one of pose tracking, where the goal is to monitor small deviations from a known pose over time, rather than figuring out the satellite's position from scratch. The PF time complexity poses another problem in the choice of it for spacecraft estimation. In particular, to have decent accuracy, the number of particles must depend exponentially to the system dimension. This is due to the phenomenon known as 'curse of dimensionality'. If the variables are independent with respect to each other, all of their probability distribution must be expressed by the same amount of particles. Moreover, since the mean and variance of the

distribution are not directly found through an analytical formula it is not simple to reproduce its results making these technique less rigorous.

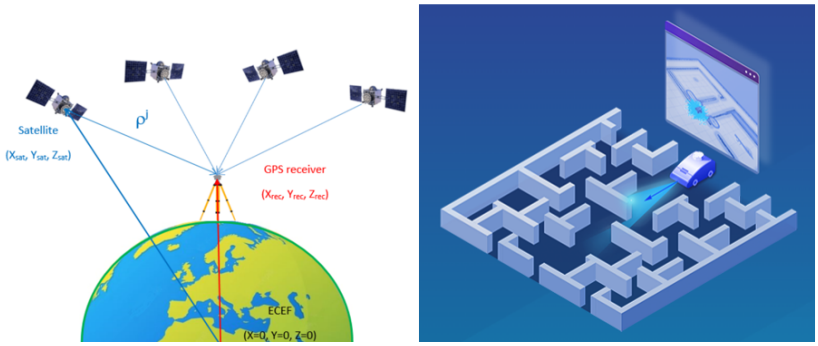

Figure 1.5: On the left a pose tracking problem in the satellite estimation scenario while on the right a global localization problems (credit: [18] and [19]).

The objective of the thesis is to estimate the states of a nonlinear system subject to non-Gaussian noise with low computational effort. To do so the UKF is chosen as a starting point in the creation of an accurate and robust filter. An advancement in the estimator performances can be given by adopting the Huber Unscented Kalman Filter (HUKF) and Huber Unscented Kalman Smoother (HUKS). The HUKF provides robustness against non-Gaussian noise, which can arise from sensor inaccuracies or disturbances in space. By incorporating Huber loss functions [20], it effectively handles outliers and ensures reliable pose estimates, maintaining accurate tracking even in uncertain conditions. On the other hand, the HUKS offers an optimized approach to trajectory estimation. While the HUKF focuses on real-time tracking, the smoother analyzes the entire trajectory, providing a more accurate global estimate over a series of observations. This makes the HUKS especially useful in docking scenarios, where refining the satellite's path relative to the docking target is critical for precise alignment. The smoother [21] ensures the trajectory is optimized, improving the final approach to the docking target.

Chapter 2

Satellite Architecture

2.1 Mission concept

The thesis reference is the PRISMA mission, led by the Swedish Space Corporation (SSC). It is a technology demonstration focused on formation flying and rendezvous technologies. The satellite system consists of two spacecraft: MAIN (the active, maneuverable satellite) and TARGET (a passive satellite with limited functionality), designed to test guidance, navigation, and control (GNC) methods for future in-orbit servicing missions. The MAIN satellite is highly maneuverable, fully equipped with six thrusters to provide torque-free translational capability. It features a 3-axis stabilization system with reaction wheels for fine attitude control, alongside GPS receivers, magnetometers and sun sensors for precision navigation. These allow MAIN to perform complex orbital maneuvers and close-range proximity operations. TARGET is a simpler spacecraft with no propulsion system but features magnetorquers for coarse 3-axis stabilization and GPS receivers for positional updates. TARGET communicates its position to MAIN via an intersatellite link (ISL), which is crucial for the rendezvous operations. PRISMA operates in a Sun-synchronous orbit (SSO), with an altitude of about 720 km x 780 km, inclination of 98.2o, Local Time of Ascending Node (LTAN) of 6:00 hours (or 18:00 hours). This ensures consistent lighting conditions for sensor operations [22].

2.2 Actuators

In a satellite, both thrusters and 3-axis stabilization system with reaction wheels are fundamental for controlling position and orientation in space. Thrusters provide high maneuverability and are essential for rapid changes in a satellite's orbit or for large-scale attitude adjustments. Instead, reaction wheels offer high-precision control without expending fuel, ideal for maintaining satellite orientation

for extended periods. By combining these two systems, satellites achieve both fine-tuned attitude control and the ability to perform orbital maneuvers.

2.2.1 Thrusters

Thrusters are small rocket engines used to adjust the satellite's position (translation) and orientation (attitude) in space. They apply a force in specific directions to change the velocity or orientation of the satellite. They are essential for tasks such as orbital adjustments, station-keeping (maintaining the satellite's correct orbit), and fine-tuning its attitude during proximity operations like docking or formation flying. The working principle focuses on expelling a propellant at high speed, generating a reaction force in the opposite direction based on Newton's Third Law of Motion. In Figure 2.1 an example of a working thruster is shown.

Figure 2.1: On the left the close-up of a thruster and on the right a working satellite thruster (credit: [23] and [24]).

Depending on the satellite's mission, the thrusters can be chemical, electric, or cold-gas thrusters. Chemical thrusters rely on the combustion of fuel, which provides high thrust and is often used for rapid orbital maneuvers. The mission in question uses high-performance green propellant (HPGP) thrusters, an environmentally friendly alternative to traditional hydrazine-based propulsion systems. Electric propulsion systems, like ion or Hall thrusters, provide much smaller thrust but with higher efficiency, making them suitable for long-duration missions requiring small adjustments over time [25]. Thrusters are often arranged in pairs or groups to provide balanced forces without introducing unwanted rotational effects. When used for attitude control, multiple thrusters are fired in specific sequences to rotate

the satellite without altering its orbital path.

2.2.2 Reaction wheels

Reaction wheels are part of a satellite's Attitude Determination and Control System (ADCS). Unlike thrusters, which apply external forces, reaction wheels manage orientation using internal forces to precisely adjust the satellite's attitude in three dimensions (3-axis). Reaction wheels consist of flywheels that spin at high speeds. When the flywheel's speed is changed, the satellite experiences an equal and opposite rotational force (due to conservation of angular momentum). This allows for fine control of the satellite's orientation without the need to expel propellant, making reaction wheels ideal for long-term missions where minimizing fuel consumption is crucial [26]. Typically, four reaction wheels are used, with three aligned along the satellite's principal axes and a fourth as a backup or for additional flexibility. Reaction wheels are primarily used for slow, precise adjustments of the satellite's orientation, which is necessary for tasks like pointing scientific instruments, solar panels, or antennas at the correct targets. They provide continuous and precise control over the satellite's orientation, unlike thrusters, which are used in short bursts. However, reaction wheels can saturate (reach their maximum speed), at which point magnetic torque rods or thrusters may be employed to desaturate them.

2.3 Sensors

Different types of sensors are employed depending on whether the goal is to measure global or relative position and orientation. For global position determination, sensors such as GPS receivers are indispensable. As said in the introduction chapter GPS provides highly accurate information about a satellite's location by triangulating signals from multiple satellites orbiting the Earth. This data is fundamental for navigating and maintaining a satellite's trajectory within its orbit. For absolute orientation (also known as attitude determination), magnetometer are widely used.

2.3.1 GPS

Using GPS for satellite navigation has become standard practice, initially developed to provide a reliable method for vehicles to determine their position autonomously. Thus, the state space vector estimated by the GPS includes the three linear variables $\mathbf{r} = [x, y, z]$. GPS works by leveraging geometric triangulation, using known satellite positions to calculate a user's position with high accuracy. To achieve this, signals from a minimum of four satellites are required, allowing the

system to determine not only the user's three-dimensional location but also the time, which is necessary to adjust for errors in the receiver's clock. GPS signals consist of pseudo-random codes (PRCs) unique to each satellite, ensuring the receiver can distinguish between them. Mathematically it is defined as the Euclidean distance between the i_{th} satellite and the receiver, i.e.;

$$\rho_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}, \tag{2.1}$$

where $\mathbf{r}_s = [x_s, y_s, z_s]$ is the position vector in ECI coordinate of the i_{th} satellite. In the context of the PRISMA mission the GPS receiver use the Coarse Acquisition (C/A) code. The C/A code is a unique sequence of pseudo-random bits used by civilian GPS receivers to determine their position. It is modulated onto the L1 carrier frequency, which operates at 1575.42 MHz. It repeats every 1,023 bits and has a chip rate of 1.023 MHz, meaning it completes one full cycle every millisecond [8]. Given the position vector $\mathbf{r}$ in ECI coordinate, the measured output $\mathbf{y}_r$ is modeled as

$$\mathbf{y}_r = \mathbf{r} + \mathbf{n}_r, \tag{2.2}$$

where $\mathbf{n}_r$ is the observation noise following a zero mean value Gaussian mixture model.

2.3.2 Magnetometer

Magnetometers play a crucial role in the determination of a satellite's orientation (or attitude) by measuring the local geomagnetic field in the satellite's body frame. The magnetometer, provides three components of the Earth's magnetic field along the satellite's body axes. In terms of mathematical modeling, this observations are related to the satellite orientation through the formula (2.3)

$$\mathbf{y}_\eta = R_{in}^b \mathbf{m}_{in} + \mathbf{n}_\eta, \tag{2.3}$$

where $\mathbf{m}_{in}$ is the known local geomagnetic field from an inertial frame whose values are obtained using models such as the International Geomagnetic Reference Field (IGRF). The IGRF models the field with respect to the distance from the centre of the Earth r, the colatitude φ which is the complementary angle of the latitude and the east longitude λ which specifies the position of a point on the surface of the Earth with respect to the Greenwich meridian. The matrix R_{in}^b rotates the inertial frame into the body frame. The overall frame rotation is given by the subsequent rotation about the z-axis of the yaw angle (ψ), about the y-axis of the pitch angle (θ) and about the x-axis of the roll angle (ϕ).

$$R_{in}^b = R_{\text{roll}}(\phi) \cdot R_{\text{pitch}}(\theta) \cdot R_{\text{yaw}}(\psi). \tag{2.4}$$

The equation (2.4) summarize the rotation about body frame axis through the composition of the three rotation matrices. The Figure 2.2 shows the local inertial frame (in this case the North-East-Down or NED frame) with respect to which the magnetic field is evaluated once the satellite cartesian coordinates are given and the body frame in which the magnetometer measurements are taken.

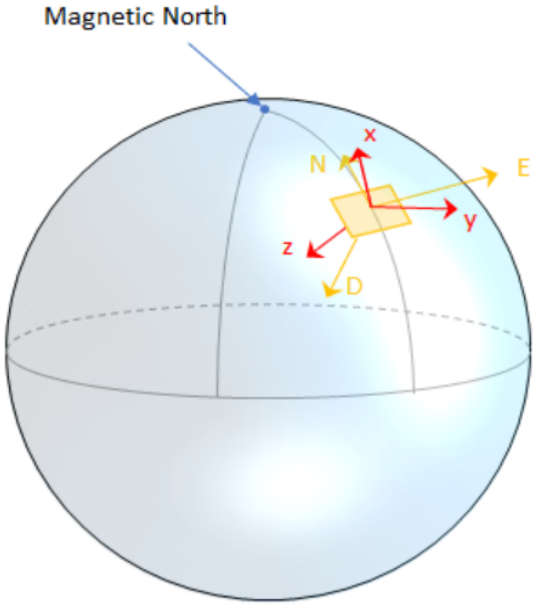

Figure 2.2: Orientation of the body frame with respect to the NED frame reconstructed through magnetometer readings (credit: [6]).

The term $\mathbf{n}_\eta$ indicates the sensor noise. It is modeled as a Gaussian mixture noise with zero mean in order to take into account all the magnetic noises that the model can not incorporate, namely the disturbances due to the spacecraft electronics and extraordinary events such as sun storms. In particular for the last one, they can produce large variance deviations from the assumed one.

Chapter 3

Sensor fusion Algorithm

3.1 System Description

Let us consider a discrete-time, nonlinear system of the form

$$\begin{aligned} \mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{w}_k) \\ \mathbf{y}_k &= h(\mathbf{x}_k, \mathbf{v}_k), \end{aligned} \tag{3.1}$$

$\mathbf{x}_k \in \mathbb{R}^n$ is the state vector, $\mathbf{w}_k \in \mathbb{R}^n$ is the process noise, $\mathbf{v}_k \in \mathbb{R}^m$ is the measurement noise, $\mathbf{y}_k \in \mathbb{R}^m$ is the vector of observed outputs, and f, h are known functions modelling the nonlinear dynamics. Both process and measurement noises are modelled at each time step k as Gaussian random variables, i.e.,

$$\mathbf{w}_k \sim \mathcal{N}(0, Q_k), \quad \mathbf{v}_k \sim \mathcal{N}(0, R_k), \tag{3.2}$$

where the covariance matrices $Q_k \succ 0$, $Q_k \in \mathbb{R}^{n \times n}$ and $R_k \succ 0$, $R_k \in \mathbb{R}^{m \times m}$ contain the noise standard deviations, that are an indicator of the state or output variable dispersion from its true value [27].

3.2 The Unscented Kalman Filter

Considering the application requirements and the processor computing capabilities, the Unscented Kalman Filter (UKF) is a middle ground in terms of accuracy and complexity. Generally speaking, the UKF filter exploits sampling points, known as sigma point, and it does not rely on the linearization of the dynamical model. Moreover, unlike PF, the number of sampling points needed to faithfully reconstruct the probability density function varies linearly with respect to the dimension of the system (see Figure 3.1). As for other filtering techniques, the UKF aims at

deriving at each time step k the best state estimate (called filtered state), given a probabilistic discrete-time dynamic model and sensor measurements [28], i.e.,

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \sim \mathcal{N}(\mathbf{x}_k|\hat{\mathbf{x}}_{k|k}, \hat{\mathbf{P}}_{k|k}), \tag{3.3}$$

where $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is the estimated probability distribution of the state variable conditioned to the measurements at time k, $\hat{\mathbf{x}}_{k|k}$ is the filtered state vector and defines the expected value of the Normal distribution, and $\hat{\mathbf{P}}_{k|k}$ is its covariance matrix.

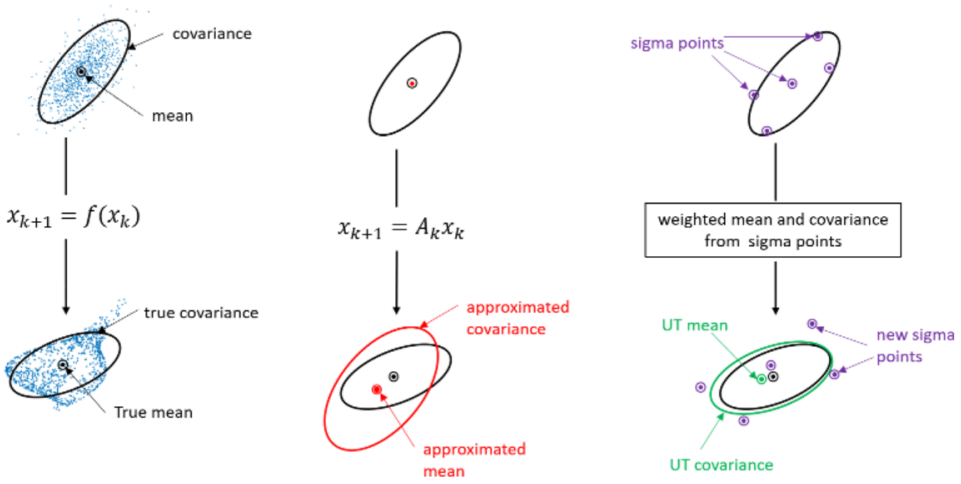

Figure 3.1: Visual confrontation between actual state propagation, EKF and UKF. The EKF, through the state transition linear approximation A_k, fails to estimate the true mean. The UKF approximate mean and covariance thanks to sigma points propagation (credit: [29]).

The objective of the UKF is to update the state estimate by propagating the prior state and measurement probability distribution through the model transition f and measurement h functions as

$$\begin{aligned} p(\mathbf{x}_{k+1}|\mathbf{x}_k) &\sim \mathcal{N}(\mathbf{x}_{k+1}|\hat{\mathbf{x}}_{k+1|k}, \hat{\mathbf{P}}_{k+1|k}) \\ p(\mathbf{y}_k|\mathbf{x}_k) &\sim \mathcal{N}(\mathbf{y}_k|\hat{\mathbf{y}}_{k|k}, \hat{\mathbf{P}}_y), \end{aligned} \tag{3.4}$$

where $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ represents the estimated probability distribution of the predicted state variable at the following time step $k+1$ conditioned to the state variable at time k, while $p(\mathbf{y}_k|\mathbf{x}_k)$ defines the the estimated probability distribution of the predicted output measurement variable at the time step k conditioned to the state vector variable at time k. With respect to EKF and PF, the problem to propagate the nonlinear dynamics to reconstruct the state posterior distribution is tackled

thanks to the unscented transform (UT). The UT is a method for computing the statistics of a random variable which undergoes a nonlinear transformation [16], exploiting (3.1), to evaluate the predicted estimate of the state $\hat{\mathbf{x}}_{k+1|k}$ and the output measurements $\hat{\mathbf{y}}_{k|k}$ as well as the predicted covariance matrices $\hat{\mathbf{P}}_{k+1|k}$ and $\hat{\mathbf{P}}_y$ of the state and output measurements, respectively. Let us consider the predicted state $\hat{\mathbf{x}}_{k|k-1}$ and covariance $\hat{\mathbf{P}}_{k|k-1}$ at time $k-1$. First, a set of $2n+1$ sigma points are generated to reconstruct the prior distribution of the true mean and covariance of the state vector, i.e.,

$$\hat{\mathbf{x}}^{(0)}_{k|k-1} = \hat{\mathbf{x}}_{k|k-1} \tag{3.5}$$

$$\hat{\mathbf{x}}^{(i)}_{k|k-1} = \hat{\mathbf{x}}_{k|k-1} + \Delta\mathbf{x}^{(i)}_p \qquad i = 1, ...2n \tag{3.6}$$

$$\Delta\mathbf{x}^{(i)}_p = \sqrt{c[\hat{\mathbf{P}}_{k|k-1}]_i}, \qquad i = 1, ..., n \tag{3.7}$$

$$\Delta\mathbf{x}^{(n+i)}_p = -\sqrt{c[\hat{\mathbf{P}}_{k|k-1}]_i}, \qquad i = 1, ..., n \tag{3.8}$$

$$c = \alpha^2(l + \kappa), \tag{3.9}$$

where the distances $\Delta\mathbf{x}^{(i)}_p$ and $\Delta\mathbf{x}^{(n+i)}_p$ of the sigma points from the estimated mean depend on the tunable parameters α and κ as well as from the i^{th} row $[\hat{\mathbf{P}}_{k|k-1}]_i$ of the predicted covariance matrix $\hat{\mathbf{P}}_{k|k-1}$. Therefore, the spread of sampling points is not constant, but varies so that the distribution profile is approximately captured. The sigma points are then propagated through the measurement function h, to obtain $2n+1$ predicted observations from every prior state as

$$\hat{\mathbf{y}}^{(i)}_{k|k} = h(\hat{\mathbf{x}}^{(i)}_{k|k-1}, \mathbf{v}_k). \qquad i = 0, ...,2n \tag{3.10}$$

The predicted observation expected value $\hat{\mathbf{y}}_{k|k}$ is obtained as the weighting sum of all the propagated estimates $\hat{\mathbf{y}}^{(i)}_{k|k}$ as

$$\hat{\mathbf{y}}_{k|k} = \sum_{i=0}^{2n} W^{(i)}_M \hat{\mathbf{y}}^{(i)}_{k|k}. \tag{3.11}$$

Then, the predicted observation covariance matrix $\hat{\mathbf{P}}_y$ is evaluated as the weighted sum of the mean square deviation between each propagated observation vector $\hat{\mathbf{y}}^{(i)}_{k|k}$, the expected value $\hat{\mathbf{y}}_{k|k}$, and the measurement noise covariance matrix R_k, i.e.,

$$\hat{\mathbf{P}}_y = \sum_{i=0}^{2n} W^{(i)}_c (\hat{\mathbf{y}}^{(i)}_{k|k} - \hat{\mathbf{y}}_{k|k})(\hat{\mathbf{y}}^{(i)}_{k|k} - \hat{\mathbf{y}}_{k|k})^\top + R_k, \tag{3.12}$$

where the term $\hat{\mathbf{P}}_y$ approximately quantifies the dispersion of the propagated sigma points with respect to the true measurement at time k. It is important to highlight

that its accuracy depends on the number of sampling points as well as on how accurate the measurement noise is modeled. Moreover, in (3.11) and (3.12), the parameters $W_M^{(i)}$ and $W_c^{(i)}$ assign a weight to each sigma point when computing the estimated mean and covariance. Specifically, for each $i = 1, \ldots, 2n$, these weighting parameters take into account the spread of the distribution through the previously defined tunable parameters α and κ other than β, as follows

$$W_M^{(0)} = 1 - \frac{n}{\alpha^2(n+\kappa)}, \qquad W_M^{(i)} = \frac{n}{2\alpha^2(l+\kappa)}, \tag{3.13a}$$

$$W_c^{(0)} = (2 - \alpha^2 + \beta) - \frac{n}{\alpha^2(n+\kappa)}, \qquad W_c^{(i)} = \frac{n}{2\alpha^2(n+\kappa)}. \tag{3.13b}$$

We want to remark that the parameter β is used to incorporate knowledge of the prior state distribution and, for a Gaussian random variable, the optimal value is typically selected as $\beta = 2$. Once $\hat{\mathbf{y}}_{k|k}$ is calculated, $\hat{\mathbf{x}}_{k+1|k}$ is found by propagating sampling points of the filtered expected value of the state $\hat{\mathbf{x}}_{k|k}$ through the UT method. The vector variable $\hat{\mathbf{x}}_{k|k}$, as well as the state vector filtered covariance matrix $\hat{\mathbf{P}}_{k|k}$, can be computed by comparing the predicted observations to the true measurements as

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\hat{\mathbf{y}}_{k|k} - \mathbf{y}_k) \tag{3.14}$$

$$\hat{\mathbf{P}}_{k|k} = \hat{\mathbf{P}}_{k|k-1} + \mathbf{K}\hat{\mathbf{P}}_y\mathbf{K}^\top, \tag{3.15}$$

where the reliability of the nonlinear measurement model with respect to the sensor sensibility (and vice versa) is quantified by an indicator known as *Kalman gain* $\mathbf{K}$, defined as

$$\mathbf{K} = \mathbf{P}_{xy}\hat{\mathbf{P}}_y^{-1}, \tag{3.16}$$

where $\mathbf{P}_{xy}$ is the state-output cross covariance matrix, which, as explained in [30], contains the covariances between all the possible couples of random variables formed by one entry of $\mathbf{x}$ and $\mathbf{y}$ i.e.,

$$\mathbf{P}_{xy} = \sum_{i=0}^{2n} W_c^{(i)}(\hat{\mathbf{x}}_{k|k-1}^{(i)} - \hat{\mathbf{x}}_{k|k-1})(\hat{\mathbf{y}}_{k|k}^{(i)} - \hat{\mathbf{y}}_{k|k})^\top. \tag{3.17}$$

In the context of state estimation, $\mathbf{P}_{xy}$ compares the dispersion of the prior estimated state distribution and the predicted observation distribution with respect to their estimated mean. Since the expected value of a distribution that represents a deterministic variable is less accurate as the standard deviation increases, $\mathbf{P}_{xy}$ is a good indicator of reliability. Hence, in the Kalman gain equation (3.16), $\hat{\mathbf{P}}_y$ serves as a regularization term. Once $\hat{\mathbf{x}}_{k|k}$ and $\hat{\mathbf{P}}_{k|k}$ are computed, the sigma points are generated at time k as in (3.5)–(3.8)

Then, the predicted expected value $\hat{\mathbf{x}}_{k+1|k}$ is calculated as the weighted mean of each posterior distribution state value $\hat{\mathbf{x}}^{(i)}_{k+1|k}$ as

$$\hat{\mathbf{x}}_{k+1|k} = \sum_{i=0}^{2n} W_M^{(i)} \hat{\mathbf{x}}^{(i)}_{k+1|k} = \sum_{i=0}^{2n} W_M^{(i)} f(\hat{\mathbf{x}}^{(i)}_{k|k}, \mathbf{w}_k), \tag{3.18}$$

whereas the predicted covariance matrix $\hat{\mathbf{P}}_{k+1|k}$ is computed as the weighted mean square deviations of each sampling point $\hat{\mathbf{x}}^{(i)}_{k+1|k}$ from the predicted expected value $\hat{\mathbf{x}}_{k+1|k}$, plus the process noise covariance matrix Q_k, i.e.,

$$\hat{\mathbf{P}}_{k+1|k} = \sum_{i=0}^{2n} W_c^{(i)} (\hat{\mathbf{x}}^{(i)}_{k+1|k} - \hat{\mathbf{x}}_{k+1|k})(\hat{\mathbf{x}}^{(i)}_{k+1|k} - \hat{\mathbf{x}}_{k+1|k})^T + Q_k. \tag{3.19}$$

Last, being the UKF a recursive filter, after updating to the time step $k+1$, the algorithm computes the new estimates from the variables predicted at the time step k, and so on, in an iterative way.

3.3 The Unscented Kalman Smoother

The purpose of a smoother is to improve the accuracy of the estimates relying on the later observations, working as a backward algorithm. With respect to the filter, the smoother operates on the state vector probabilistic distribution $p(\mathbf{x}_k|\mathbf{y}_{1:T})$ at time step $k \in [1, T]$ conditioned to the current and future measurements and filtered state estimates from 1 to T, where T is the length of the time interval in which the algorithm is applied, such that

$$p(\mathbf{x}_k|\mathbf{y}_{1:T}) \sim \mathcal{N}(x_k|\mathbf{x}^s_k, \mathbf{P}^s_k). \tag{3.20}$$

The knowledge of future observations, expressed in (3.20) using the Bayesian statistics notation, allows to define the most plausible state vector $\mathbf{x}^s_k$ so that in the following steps those measurements are obtained (if reliable). Thus, it shrinks the state vector probabilistic dispersion, quantified by the smoothing covariance matrix $\mathbf{P}^s_k$. The closed form solution to this problem is given by the Rauch-Tung-Striebel algorithm [28] also called Kalman smoother. Its working principle is visually shown in Figure 3.2.
The unscented Kalman smoother (UKS) is a Gaussian approximation where the non-linearity is approximated using the unscented transform described in Section 3.2. First, the length T of the time intervals in which the smoother operates is defined as

$$T = \frac{N_{tot}}{n_{cycles}}, \tag{3.21}$$

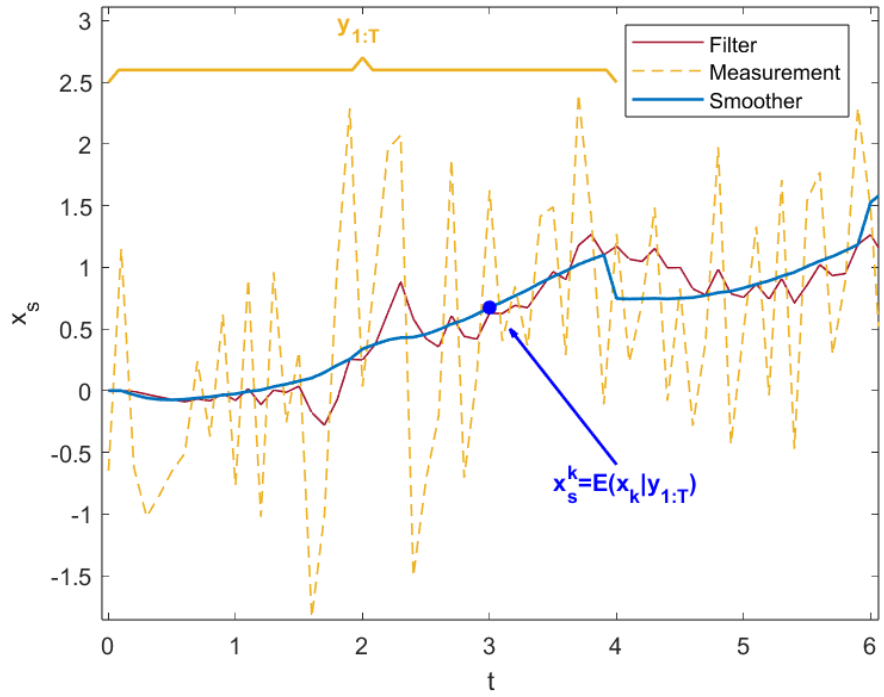

Figure 3.2: The graph shows the number of past and future measurements ($\mathbf{y}_{1:T}$) used to smooth out ($\mathbf{x}_s^k$) the filtered estimate inside that interval.

where N_{tot} is the total number of time steps for which measurement data are available and n_{cycles} are the number of intervals. It is important to remark that a lower number of intervals improves the smoothness action, but it increases the computational weight, while increasing the interval amount means reducing the smoother impact on the filter. Then, the filtered state vector estimate is compared with the difference between the predicted $\hat{\mathbf{x}}_{k+1|k}$ and the smoothed state vector $\mathbf{x}_{k+1}^s$ at time step $k+1$. To smooth the path, the $k+1$ step difference must be the same of the k step, such that $\mathbf{x}_k^s$ is the smoothed state vector estimate is given by

$$\mathbf{x}_k^s = \hat{\mathbf{x}}_{k|k} + \mathbf{G}(\mathbf{x}_{k+1}^s - \hat{\mathbf{x}}_{k+1|k}). \tag{3.22}$$

where the gain $\mathbf{G}$ accounts for the predicted variables reliability and it is defined as the ratio between the cross-covariance matrix $\mathbf{P}_{cv}$ (see section 3.2 and reference [30]) and $\hat{\mathbf{P}}_{k+1|k}$, i.e.

$$\mathbf{G} = \mathbf{P}_{cv}\hat{\mathbf{P}}_{k+1|k}^{-1}. \tag{3.23}$$

Specifically, G_t is the smoother equivalent of the Kalman gain, and it shows how much the smoothed state deviates from the real behaviour. When the predicted

covariance matrix increases, G_t decreases and so it does the error term weight. It means that the uncertainty introduced by the process noise is too high to be handled by the smoother. Analogous considerations can be made for the smoothed covariance matrix $\mathbf{P}_k^s$ at time k

$$\mathbf{P}_k^s = \hat{\mathbf{P}}_{k|k} + \mathbf{G}(\mathbf{P}_{k+1}^s - \hat{\mathbf{P}}_{k+1|k})\mathbf{G}^\top, \tag{3.24}$$

where $\mathbf{P}_{xy}$ compares the filtered $\hat{\mathbf{x}}_{k|k}$ and propagated $\hat{\mathbf{x}}_{k+1|k}$ state vector dispersion with respect to their variable mean (3.17). It quantifies the increase in standard deviation when propagating the state estimate $\hat{\mathbf{x}}_{k|k}$ through the modeled true state transition function. Thus, it is an indicator of the system dynamics reliability. On the other hand, the matrix $\hat{\mathbf{P}}_{k+1|k}$ regularize the smoother gain $\mathbf{G}$.

3.4 The Huber-based approach

Let's recall the main principle of the Kalman filter. The Kalman filter is a recursive dynamic least square algorithm. It is based on the minimization of the sum of the squared difference between the i_{th} measured datum y_i from a set of n data corrupted by noise and a model $\hat{y}_i = h(x_i)$

$$J(\mathbf{x}) = \sum_{i=1}^{n}(e_i)^2 = \sum_{i=1}^{n}(y_i - h(x_i))^2, \tag{3.25}$$

where the difference e_i is also called the residual of the i_{th} observation y_i and $J(\mathbf{x})$ represents the cost function to derive in order to obtain the optimal estimator $\mathbf{x}$ that minimizes all the residuals [31]. Assuming data points y_i are corrupted by Gaussian noise the algorithm doesn't have bias and has minimum variance among all estimators. However, real-world data often deviates from these ideal conditions (see Figure 3.3), leading to significant limitations in the classical Kalman filter's performance. These assumptions, while reasonable in many scenarios, contribute to the filter's sensitivity to contaminated distributions.
The so-called outliers, are data points that deviate significantly from the expected distribution of the data. They can arise due to various reasons, such as sensor malfunctions, environmental changes, or data entry errors. Since their residual is much greater than usual, their behaviour is enhanced by the mean squared metrics. Thus, they can significantly compromise the state estimates because the filter gives equal weight to all measurements, including anomalous ones. The objective of this and the next sections is to derive a robust algorithm against uncertain measurements. The concept of robustness relates to the level of perturbation that the estimator can handle before breaking down completely [21], for example due to contamination of the assumed distribution with another one with different

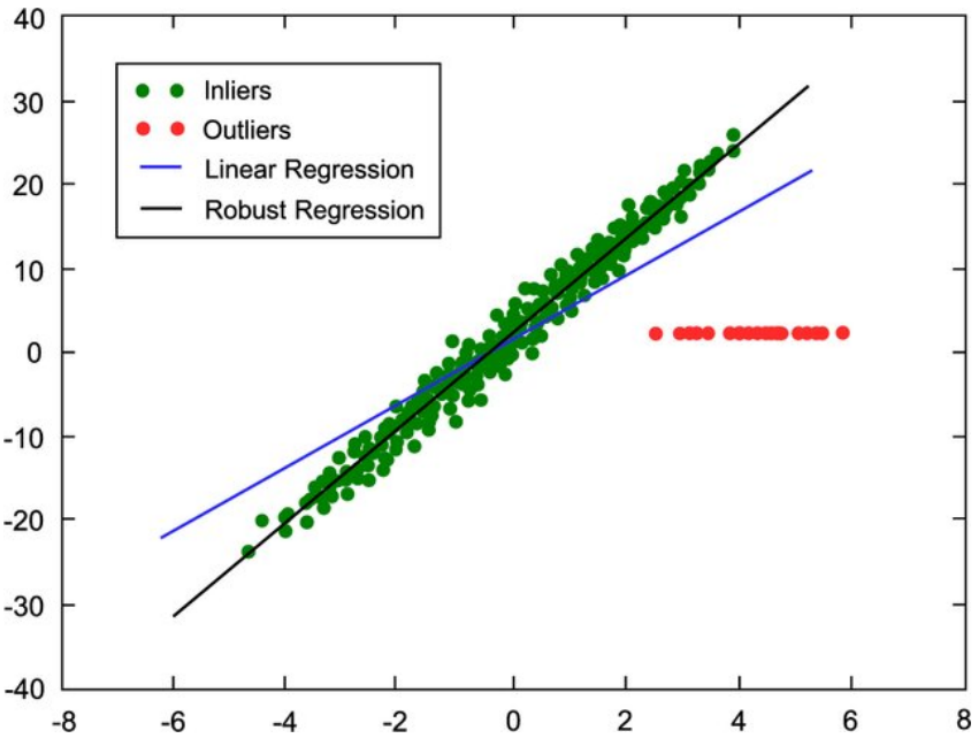

Figure 3.3: Behaviour of Least Square algorithm for a linear regression problem in presence of anomalous measurements. The angular coefficient choice is biased and deviates from the optimal value for the majority of points. (credit: [20]).

characteristics. An approach to increase robustness is to replace the standard cost with the one defined by the Huber approach and combine the l_2 and l_1 norms ([ref])

$$J(\mathbf{x}) = \sum_{i=1}^{n} \rho(e_i), \tag{3.26}$$

where the function ρ behaves like this

$$\begin{cases} \rho(e_i) = (y_i - h(x_i))^2 & if \quad e_i \leq \gamma \\ \rho(e_i) = |y_i - h(x_i)| & if \quad e_i > \gamma \end{cases} . \tag{3.27}$$

The constant γ is a threshold parameter. When e_i is lower than γ, the cost function is a quadratic function. When e_i is bigger than γ it behaves linearly. It ensures that smaller residuals (which are likely to be part of the main distribution) are treated with greater importance while larger residuals (potential outliers) are down weighted. The robustness effect depends on the choice of γ:

- $\gamma \to 0$: the robustness effect increases as more observations are treated as probable outliers;
- $\gamma \to \infty$: the Huber function behaves like a least square minimization algorithm.

If we derive the Huber cost function and we divide by the residual e_i, i.e.,

$$\begin{cases} \psi(e_i) = 1 & if \quad e_i \leq \gamma \\ \psi(e_i) = \gamma/|e_i| & if \quad e_i > \gamma \end{cases}, \tag{3.28}$$

we obtain the weights ψ_i which address the uncertainty of the measurement by modifying its assumed variance σ_i^2 [32]. As the residual e_i grows larger than γ so it does its variance as it takes into account the probability that the measurement is an outlier. Vice versa, for a value of e_i smaller than γ the observation is expected to follow the assumed distribution and the variance remains unchanged.

3.5 Huber-based Unscented Kalman Filter

The notion of measurement noise uncertainty due to contamination of a different Gaussian distribution from the assumed one, can be incorporated in the UKF algorithm by modifying the measurement noise covariance matrix R_k. The procedure is the following. During the measurement update step, the predicted observation $\hat{\mathbf{y}}_{k|k}$ defined in (3.12) is computed as usual. Then, for each entry of the vector $\hat{\mathbf{y}}_{k|k}$ the corresponding measurement residual ζ_j is evaluated as

$$\zeta_j = |\mathbf{y}_k(j) - \hat{\mathbf{y}}_{k|k}(j)|. \tag{3.29}$$

where $\mathbf{y}_k(j)$ represents the j_{th} component of the observation vector $\mathbf{y}_k$. If the j_{th} residual is larger than the value of the threshold γ, then the variance of the respective measurement is modified through the weight

$$\psi_j = \gamma/|\zeta_j|, \tag{3.30}$$

to guarantee robustness in case the measurement itself represents an outlier. Otherwise, the variance remains the same since the weight ψ_i is equal to 1. In this case the algorithm falls back to the classical UKF structure. The diagonal matrix of the weights Ψ is computed. The structure of matrix R_k is reshaped as

$$\tilde{R_k} = {R_k}^{1/2} \Psi^{-1} {R_k}^{T/2}, \tag{3.31}$$

where Ψ is a diagonal matrix with dimension equal those of $\mathbf{y}_k$ and entries $\psi_1, ..., \psi_n$. Thus, the measurement covariance matrix $\tilde{\mathbf{P}}_y$ is evaluated as

$$\tilde{\mathbf{P}}_y = \sum_{i=0}^{2n} W_c^{(i)} (\hat{\mathbf{y}}_{k|k}^{(i)} - \hat{\mathbf{y}}_{k|k})(\hat{\mathbf{y}}_{k|k}^{(i)} - \hat{\mathbf{y}}_{k|k})^\top + \tilde{R}_k. \tag{3.32}$$

The new Kalman gain $\tilde{\mathbf{K}}$ is defined as the ratio between the state-output cross covariance matrix $\mathbf{P}_{xy}$ (defined in 3.17) and the measured covariance matrix $\tilde{\mathbf{P}}_y$

$$\tilde{\mathbf{K}} = \mathbf{P}_{xy} \tilde{\mathbf{P}}_y^{-1}, \tag{3.33}$$

$\tilde{\mathbf{K}}$ is then used to obtain the filtered state $\hat{\mathbf{x}}_{k|k}$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \tilde{\mathbf{K}}(\hat{\mathbf{y}}_{k|k} - \mathbf{y}_k) \tag{3.34}$$

Equations (3.33) and (3.34) illustrate how the Huber-based filter modifies the Kalman gain (and consequently the filtered state) to incorporate robustness. Particularly, Huber's weight function assigns lower weights to measurements with large residuals (i.e., potential outliers). Thus, $\tilde{\mathbf{P}}_y$ will increase and $\tilde{\mathbf{K}}$ will decrease. This means that outliers have less influence on the state update, preventing them from altering the estimate significantly. In contrast, the classic Kalman gain treats all residuals equally, making it more sensitive to the effects of outliers. Moreover the introduction of the Huber's weight function allows for a more flexible response. For small residuals, where the data likely follows the Gaussian assumption, the filter behaves similarly to the classic Kalman filter. For large residuals, as previously said, it shifts towards a more conservative approach, reducing the impact of these measurements. This adaptiveness ensures a better balance between sensitivity and robustness.

3.6 Huber-based Unscented Kalman Smoother

The objective of any Huber-based approach is to reduce the contribution of uncertain measurements in the computation of the estimator. However, this results in the increment of the estimator variance. A Huber robust smoother improves the trajectory by considering not only past and present data but also future observations reducing statistical uncertainty and variance. Using both a Huber robust filter and a Huber robust smoother ensures consistent robustness throughout the estimation process without sacrificing estimator certainty. Given each j_{th} entry of $\mathbf{x}_{k+1}^s$, the corresponding residual ξ_j is computed as

$$\xi_j = |\mathbf{x}_{k+1}^s(j) - x_{k+1|k}(j))|. \tag{3.35}$$

where $x_{k+1|k}(j)$ is the j_{th} component of the model predicted state at time step k+1. If ξ_j is higher than the threshold γ, the smoothness action deviates too much from

the estimate of the future state. The result is an increase in the variance of the j_{th} smoothed variable. Meanwhile, if ξ_j is lower than γ the smoother behave as a classical UKS

$$\tilde{Q}_k = Q^{1/2}\Xi^{-1}Q^{\top/2}, \tag{3.36}$$

where the matrix Ξ is diagonal of dimension coherent with the state vector and entries $\xi_1, ...\xi_n$ and the updated covariance matrix is given by

$$\hat{\mathbf{P}}_{k+1|k} = \sum_{i=0}^{2n} W_c^{(i)}(\hat{\mathbf{x}}_{k+1|k}^{(i)} - \hat{\mathbf{x}}_{k+1|k})(\hat{\mathbf{x}}_{k+1|k}^{(i)} - \hat{\mathbf{x}}_{k+1|k})^T + \tilde{Q}_k. \tag{3.37}$$

The new smoother gain $\tilde{\mathbf{G}}$ [21] is defined as the ratio between the cross covariance matrix $\mathbf{P}_{cv}$ and the updated covariance matrix $\tilde{\mathbf{P}}_{k+1|k}$

$$\tilde{\mathbf{G}} = \mathbf{P}_{cv}\tilde{\mathbf{P}}_{k+1|k}^{-1}, \tag{3.38}$$

and the robust update of the covariance matrix smoother $\tilde{\mathbf{P}}_{k+1|k}$ allows for a change in $\tilde{\mathbf{G}}$ so that it reduces in the case of an high deviation of the smoothed state $\mathbf{x}_{k+1}^s$ from the predicted state $\mathbf{x}_{k+1|k}$.

$$\hat{\mathbf{x}}_k^s = \hat{\mathbf{x}}_{k|k} + \tilde{\mathbf{G}}(\mathbf{x}_{k+1}^s - \mathbf{x}_{k+1|k}). \tag{3.39}$$

This implies that in the case of an high deviation of the smoothed state $\mathbf{x}_{k+1}^s$ from the predicted state $\mathbf{x}_{k+1|k}$, $\tilde{\mathbf{G}}$ and the smoother action decrease.

3.7 Accuracy and uncertainty

Accuracy and uncertainty are the indicators of an estimator's performance. **Accuracy** measures how close the estimated state is to the true state of the system. The **uncertainty** represents the estimator confidence in producing a certain estimate [31]. It is an indicator of the measurement variability. When the noise affecting the system is Gaussian, the UKF can produce accurate state estimates with a well-defined uncertainty. Outliers and non-Gaussian noise can lead to biased state estimates, as the UKF weights all measurements equally, regardless of their consistency with the predicted state. The Huber Unscented Kalman Filter addresses the limitations of the UKF by introducing robustness to non-Gaussian noise. As illustrated in the last two sections, the key innovation in the HUKF lies in the application of the Huber loss function, which distinguishes between small and large residuals during the update step. This allows the HUKF to reduce the influence of outliers by assigning them a lower weight compared to measurements that are closer to the predicted state. As the variability of the measurements increases so it does

the estimate of the standard deviation. Let us consider a Gaussian mixture model composed of two Normal distributions with zero mean and variance respectively $\sigma_1 = 1$ and $\sigma_2 = 10$, where the filter assumed noise variance is $\sigma_R = 1$. The two graphs presented in Figure 3.4 show the accuracy and uncertainty behaviour of the estimator as the threshold parameter γ and the distribution probability parameter p change.

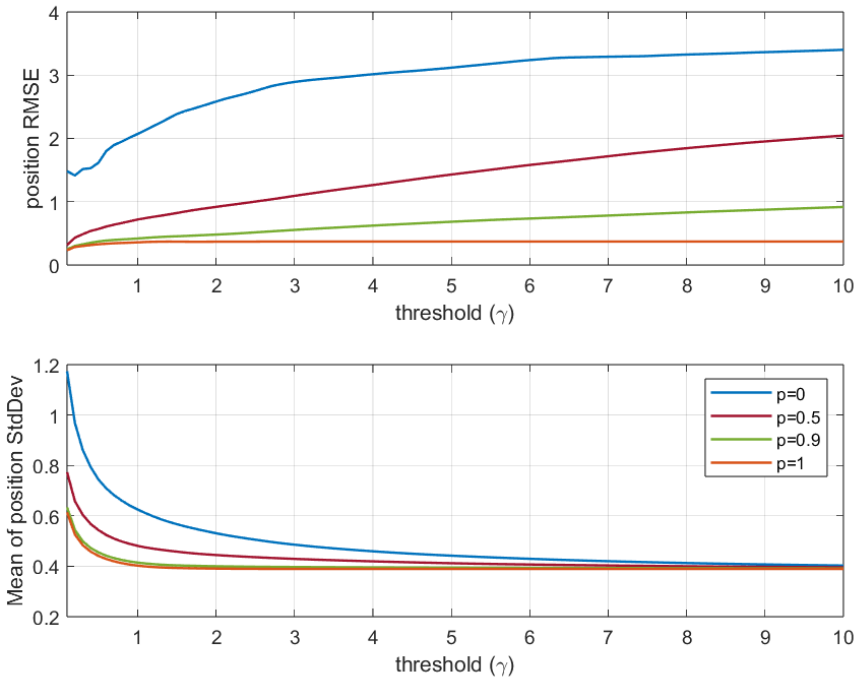

Figure 3.4: position RMSE and standard deviation of Huber-based UKF for $p = [0,0.5,0.9,1]$ and with respect to γ in the interval [0.1,10].

As gamma increases, the Root Mean Square Error between estimate and true value (RMSE, a measure of accuracy) decreases and the standard deviation increases. This is expected because robust estimators are designed to reduce the impact of outliers, which generally leads to more accurate predictions (as indicated by a lower RMSE). The goal is to find the optimal threshold gamma to minimize RMSE without inflating uncertainty excessively. This compromise ensures that the system remains reliable while not being overly conservative in its estimates. For probability p=1 the best value for the threshold is $\gamma = 1.345$. It is the value of γ at which approximately both standard deviation and RMSE stabilize. Basically, it is calculated as the mean value of the two parameters derivatives with respect to γ. The importance of achieving the perfect balance between accuracy and uncertainty

can be seen also when confronting distributions with different p. As the probability approaches 1, the distribution is close to the Normal one with lower variance. Since it matches the filter assumptions, the accuracy improves. As the distribution deviates from the assumed noise model (p=0), the accuracy worsens. Thus, if the uncertainty is not adequately accounted for, the system confidence may be too high even when the actual noise differs significantly from the predicted behavior. Managing uncertainty ensures that even in these extreme scenarios, the system remains secure.

Chapter 4

Orbital Mechanics

4.1 Introduction

The scope of the thesis is to estimate the pose over time of a satellite in orbit around the Earth. The position is obtained by solving the Keplerian laws of motion for a two body problem and the orientation through the Euler equations. Before modeling the dynamics, the chapter analyzes the reference frames in which the variables are depicted.

4.2 Reference frames

A reference frame $\mathcal{F}$ in three dimension is uniquely defined by a basis of three orthonormal vectors $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ and an origin $\mathcal{O}(x, y, z)$. Any three dimensional vector $\mathbf{x}$ can be described as a linear combination of the orthonormal basis $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$.

$$\mathbf{x} = \sum_{i=1}^{3} x_i \mathbf{e}_i \tag{4.1}$$

The same vector can be expressed in another frame $\mathcal{F}'$ with basis $\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3$

$$\mathbf{x} = \sum_{i=1}^{3} x'_i \mathbf{e}'_i \tag{4.2}$$

Given $\mathbf{x}$ expressed in $\mathcal{F}$, the component in $\mathcal{F}'$ is obtained by taking the dot product of the basis vector and the vector $\mathbf{x}$.

$$x'_k = \mathbf{e}'_k \cdot \mathbf{x} = \sum_{j=1}^{3} (\mathbf{e}'_k \cdot \mathbf{e}_j) x_j \tag{4.3}$$

If the same is done for all the other components of $\mathbf{x}$, we get the matrix $D_{F'F}$ of transformation between the two frames $\mathcal{F}$ and $\mathcal{F}'$.

$$D_{F'F} = \begin{pmatrix} \mathbf{e}_1 \cdot \mathbf{e}'_1 & \mathbf{e}_1 \cdot \mathbf{e}'_2 & \mathbf{e}_1 \cdot \mathbf{e}'_3 \\ \mathbf{e}_2 \cdot \mathbf{e}'_1 & \mathbf{e}_2 \cdot \mathbf{e}'_2 & \mathbf{e}_2 \cdot \mathbf{e}'_3 \\ \mathbf{e}_3 \cdot \mathbf{e}'_1 & \mathbf{e}_3 \cdot \mathbf{e}'_2 & \mathbf{e}_3 \cdot \mathbf{e}'_3 \end{pmatrix} \tag{4.4}$$

$D_{F'F}$ as direction cosine matrix (DCM) because their elements are the cosines of the angles between the basis vectors in the two reference frames. In order to obtain the opposite frame transformation it is sufficient to transpose matrix $D_{F'F}$.

$$D_{FF'} = D^T_{F'F} \tag{4.5}$$

4.3 The Perifocal Reference Frame

To accurately describe the trajectory of a satellite traveling around the Earth, it is essential to first construct the orbital model. This process involves defining a set of parameters and a suitable coordinate system directly associated with the satellite's orbit called perifocal coordinate system.

4.3.1 Key parameters shaping the orbit

The following parameters are the fundamental elements that define the shape, and size of the orbit. The **True Anomaly** (ν) is the angle between the direction of the perigee and the current position of the satellite as it moves along its orbit. In the perifocal frame, the true anomaly is measured directly from the P-axis, making it simple to calculate the satellite's position at any point in its orbit. The **Eccentricity** (e) defines the shape of the orbit in terms of how elliptical it is. The term goes from 0 (circular orbit) to 1. Instead, the **Semi-Major Axis** (a) defines the size of the orbit. It is the longest diameter of the elliptical orbit. When it comes to extreme parameters, the **Radius of Perigee** (r_p) is the distance from the center of the Earth to the satellite when it is at perigee, the point where the satellite is closest to Earth. It can be calculated as the sum of the Earth's mean radius and the altitude of the satellite at perigee, which is described by our mission as to be 720 km, i.e.;

$$r_p = R_{Earth} + h_p = 6371km + 720km = 7091km. \tag{4.6}$$

while the **Radius of Apogee** (r_p) is the distance from the center of the Earth to the satellite when it is at apogee, the point where the satellite is farthest from Earth [33]. It is similarly calculated as the Earth's mean radius plus the satellite's altitude at apogee, which our mission says is 780 km, i.e.;

$$r_a = R_{Earth} + h_a = 6371km + 780km = 7151km. \tag{4.7}$$

Given the radius of perigee and apogee, the semi-major axis and the eccentricity determining respectively the orbital size and shape can be computed. The semi-major axis, which is the average of the perigee and apogee distances, is calculated as:

$$a = \frac{r_p + r_a}{2}, \tag{4.8}$$

where r_a and r_p are respectively the apogee and perigee radii. The formula for eccentricity is given by:

$$e = \frac{r_a - r_p}{r_a + r_p}. \tag{4.9}$$

The numerator $r_a - r_p$ represents the difference in distance between the apogee and perigee. This difference reflects the "stretch" of the orbit or how far apart the closest and farthest points are from the Earth's center. The denominator $r_a + r_p$ represents the total distance spanned by the orbit along the major axis. By dividing the distance difference $r_a - r_p$ by this total distance, the formula normalizes the stretch of the orbit relative to its size.

4.3.2 Defining the perifocal orthonormal basis

The perifocal coordinate system [33] is specifically aligned with the satellite's orbital plane, making it an ideal starting point for orbit construction. Its origin is at the focus of the elliptical orbit. The right-handed coordinate system is formed by these three axes. The **P-axis** points directly towards the periapsis (or perigee), the point in the orbit where the satellite is closest to the Earth. It lies along the semi-major axis of the orbit. The **Q-axis** is perpendicular to the P-axis and still within the orbital plane, the Q-axis points in the direction of the satellite's motion at perigee. Finally, the **W-axis** is perpendicular to the orbital plane, pointing in the direction of the orbit's angular momentum vector ($\vec{h}$). This axis is essential for defining the plane in which the satellite moves. The Figure 4.1 highlights the P and Q axes that lie within the satellite's orbital plane. The P-axis, which points towards the perigee, and the Q-axis, perpendicular to it within the plane, define the satellite's position vector ($\mathbf{r}_{\mathrm{PQW}}$). Although not visible in the image, the W-axis completes this right-handed coordinate system by extending perpendicular to the orbital plane. The true anomaly, denoted as ν, represents the angle between the P-axis and $\mathbf{r}_{\mathrm{PQW}}$ (the satellite's current position vector).

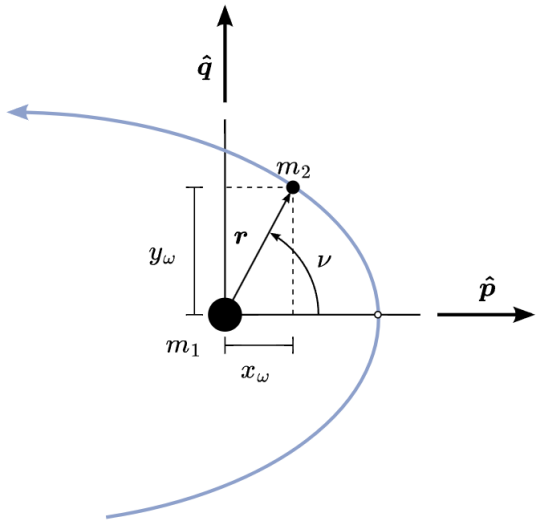

Figure 4.1: Orbit representation in PQW coordinate system (credit: [34]).

4.3.3 Establishing the Satellite's position in Perifocal Coordinates

With the semi-major axis and the eccentricity determined, the satellite's position at any point in its orbit can be expressed using the true anomaly ν in the perifocal frame. The position vector in perifocal coordinates can be described as:

$$\mathbf{r}_{\mathrm{PQW}} = \frac{a(1-e^2)}{1+e\cos v}\begin{pmatrix}\cos v\\ \sin v\\ 0\end{pmatrix}. \tag{4.10}$$

If the satellite's orbit starts exactly at perigee, the formula simplifies to:

$$\mathbf{r}^{in}_{\mathrm{PQW}} = \frac{a(1-e^2)}{1+e\cos v}\begin{pmatrix}1\\ 0\\ 0\end{pmatrix}, \tag{4.11}$$

where $\mathbf{r}^{in}_{\mathrm{PQW}}$ represents the position initial condition. This equation reflects the fact that, at the first time instant, the angle ν is 0 ($cos(\nu) = 1$ and $sin(\nu) = 0$).

4.4 The Earth Centered Inertial (ECI) frame

The Earth-Centered Inertial (ECI) frame is a fundamental reference frame in orbital mechanics. It is an inertial frame, meaning it does not rotate with respect to the distant stars, and is non-accelerating. This frame is a standard for describing the satellite's motion.

4.4.1 ECI orthonormal basis

The ECI frame [8] is centered at the Earth's center of mass. The orthonormal basis is formed by the **X-axis** that points towards the vernal equinox (a fixed direction in space), the **Z-axis** is directed along the Earth's rotation axis towards the North Pole and the **Y-axis** completes the right-handed system, lying in the Earth's equatorial plane.

4.4.2 Transformation from PQW to ECI frame

Since the P and Q axes of the perifocal coordinate system lie within the orbital plane of the satellite, the perifocal system is intrinsically linked to the orientation of the orbit relative to the celestial sphere. This means that the perifocal frame itself does not provide a global reference for the satellite's position and trajectory in relation to the fixed stars as opposed to the ECI coordinate system. To transform from the perifocal coordinate system to the ECI frame, we use three orbital elements, known as Keplerian elements [33]. The **Inclination** (i) is the angle between the orbital plane and the Earth's equatorial plane. This angle defines the tilt of the orbit relative to the Earth's equator ($i = 98.2°$). The **Right Ascension of the Ascending Node** (**RAAN**, Ω) is the angle measured in the equatorial plane from the vernal equinox to the ascending node, which is the point where the satellite crosses the equatorial plane from south to north ($\Omega = 155.85°$). The **Argument of Periapsis** (ω) is the angle within the orbital plane from the ascending node to the periapsis, measured in the direction of the satellite's motion ($\omega = 0°$). These angles describe the orbit orientation with respect to the fixed stars (Figure 4.2).
The transformation from the perifocal coordinate system (PQW) to the ECI frame is the result of three successive rotations. First, it is required a rotation by Ω about the Z-axis (from the inertial frame) to align the ascending node with the x-axis, i.e.;

$$R_1(\Omega) = \begin{pmatrix} \cos\Omega & -\sin\Omega & 0 \\ \sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{4.12}$$

then, a rotation by i about the X-axis (aligned with the line of nodes) to tilt the orbital plane relative to the equatorial plane, i.e.;

$$R_2(i) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{pmatrix}, \tag{4.13}$$

finally a rotation by ω about the Z-axis (now aligned with the line of nodes) to position the periapsis in the orbital plane, i.e.;

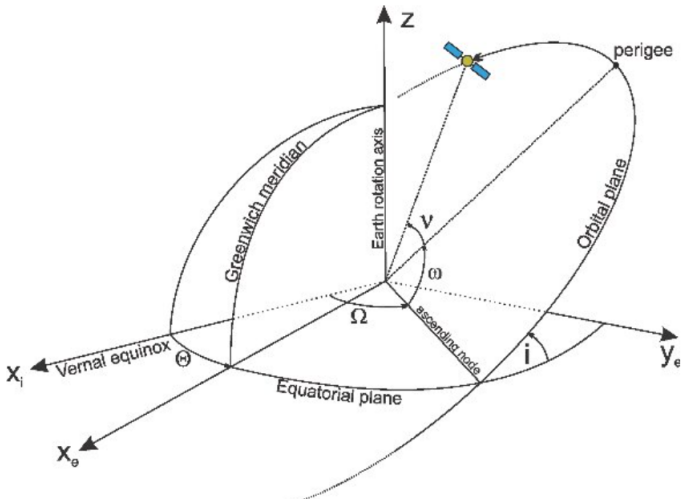

Figure 4.2: Frame transformation from PQW to ECI through the Keplerian elements i, Ω, ω (credit: [35]).

$$R_3(\omega) = \begin{pmatrix} \cos\omega & -\sin\omega & 0 \\ \sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.14}$$

The overall transformation matrix is the product of these three individual rotation matrices:

$$\mathbf{R}_{\mathrm{ECI}}^{\mathrm{PQW}} = R_1(\Omega)R_2(i)R_3(\omega). \tag{4.15}$$

The position vector in ECI coordinates can thus be derived from the equivalent in the perifocal frame: (equation)

$$\mathbf{r}_{\mathrm{ECI}} = \mathbf{R}_{\mathrm{ECI}}^{\mathrm{PQW}}\mathbf{r}_{\mathrm{PQW}}. \tag{4.16}$$

4.5 The two-body problem

The two-body problem [8] is a classical problem in celestial mechanics that describes the motion of two bodies interacting through gravity. In the context of a satellite orbiting the Earth, this problem simplifies to modeling the satellite's orbit around a central mass. The gravitational force between two masses m (satellite) and $M_E = 5.972 \times 10^{24}\,\mathrm{kg}$ (Earth) is given by:

$$\mathbf{F} = -\frac{GM_E m}{|\mathbf{r}_{\mathrm{ECI}}|^3}\mathbf{r}_{\mathrm{ECI}}, \tag{4.17}$$

where, $G = 6.6743 \times 10^{-11}\,\mathrm{m^3\,kg^{-1}\,s^{-2}}$ is the gravitational constant and $\mathbf{r}_{\mathrm{ECI}}$ is the position vector of the satellite relative to the center of the Earth in the ECI reference system. Applying Newton's second law to the satellite, we get:

$$m\frac{d^2\mathbf{r}_{\mathrm{ECI}}}{dt^2} = -\frac{GMm}{|\mathbf{r}_{\mathrm{ECI}}|^3}\mathbf{r}_{\mathrm{ECI}}, \tag{4.18}$$

simplifying, the differential equation governing the satellite's motion is:

$$\frac{d^2\mathbf{r}_{\mathrm{ECI}}}{dt^2} = -\frac{GM}{|\mathbf{r}_{\mathrm{ECI}}|^3}\mathbf{r}_{\mathrm{ECI}}. \tag{4.19}$$

4.6 Local-Vertical-Local-Horizontal frame

The Local Vertical, Local Horizontal (LVLH) coordinate system is a reference frame [8] that is commonly used in satellite operations, particularly for tasks that require precise alignment and coordination between multiple spacecraft, such as refueling, docking, or maintenance missions.

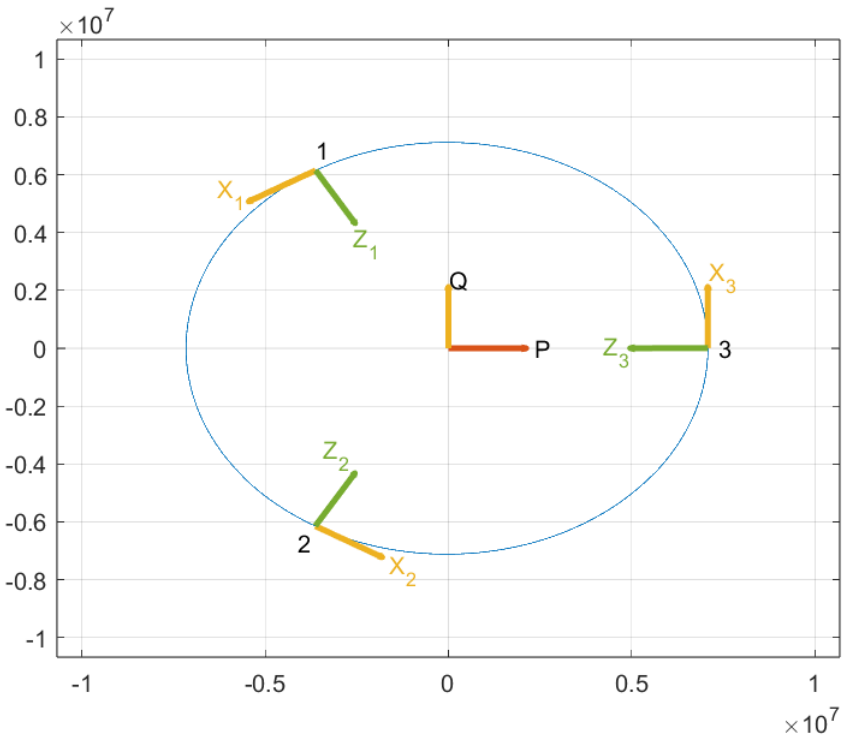

Figure 4.3: Representation in the perifocal system of the LVLH frame displacement across three different orbital points.

The LVLH coordinate system is tied to the orbital motion of the satellite, (meaning it moves along with the satellite as it travels around the Earth). Figure 4.3 and 4.4 show the LVLH orthonormal basis in three different instants of the satellite revolution around the Earth, respectively in PQW and ECI reference system.

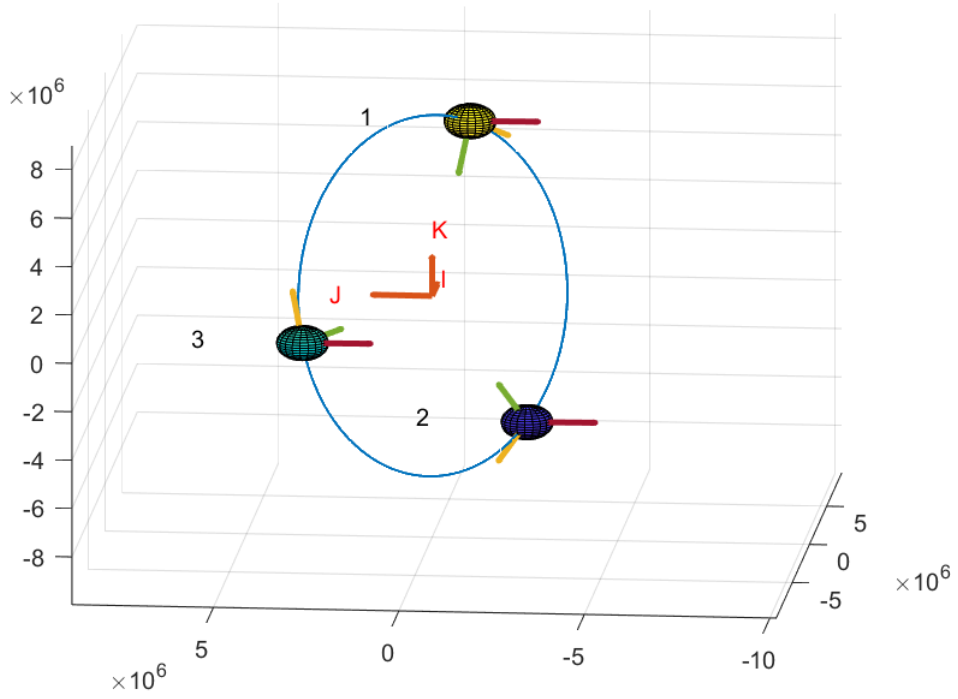

Figure 4.4: Representation in the ECI system of the LVLH frame displacement across three different orbital points.

When two satellites are interacting, the body frame of the chaser satellite (the one performing the operation) must be accurately aligned with the LVLH frame of the target satellite to accomplish the task. The LVLH coordinate system axes are oriented as follows. The $\mathbf{X_L - axis}$ points in the direction of the satellite's velocity vector as it moves along its orbit. This means it is tangential to the orbit at the satellite's current position and directed forward along the satellite's path. The $\mathbf{Y_L - axis}$ is perpendicular to the orbital plane. It points in the direction of the orbital angular momentum vector. The $\mathbf{Z_L - axis}$ points along the nadir vector towards the center of the orbital focus.

4.7 Body Frame

The body frame of a satellite is a non-inertial coordinate system that is fixed relative to the satellite itself, moving and rotating as the satellite does. This system is centered at the satellite's center of mass, and its axes are aligned with the principal axes of the satellite's structure (Figure 4.5). The rotations around the axes of its body frame are described by the Euler angles [8]. Namely, **Yaw** (ψ) is the rotation around the Z-axis of the body frame. This angle changes the direction the satellite is facing horizontally. The **Pitch** (θ) is the rotation around the Y-axis and it adjusts the satellite's nose up or down, similar to a pitching motion. The **Roll** (ϕ) is the rotation around the X-axis which tilts the satellite side-to-side, as in rolling along the direction of motion.

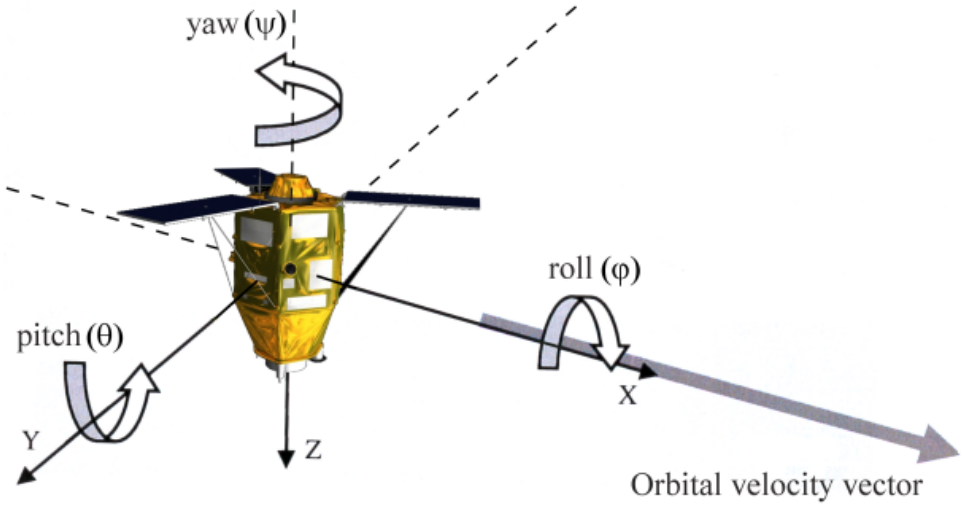

Figure 4.5: Representation of the body reference system, with its axes and angles, as it is aligned with the LVLH frame (credit: [36]).

4.7.1 Body angular velocity description through Euler angles

The angular velocity vector of the body reference system relative to the LVLH frame can be expressed through the derivatives of EUler angles. First, we need to express through Euler angles the rotation matrix form the body frame to the LVLH system. We used the 3-2-1 rotation method, known as yaw-pitch-roll

$$R_{\mathrm{b}}^{LVLH} = R_{x''}(\phi)R_{y'}(\theta)R_{z}(\psi), \tag{4.20}$$

The first step is a rotation by the yaw angle ψ around the Z-axis of the body frame. This rotation changes the orientation of the satellite about its vertical axis. After this rotation, the new intermediate X' and Y' axes are obtained, i.e.;

$$R_z(\psi) = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4.21}$$

Then, it occurs a rotation by the pitch angle θ around the new Y'-axis (the axis after the first rotation). This rotation tilts the satellite up or down relative to its initial horizontal plane. After this rotation, new intermediate axes, X" and Z", are established, i.e.;

$$R_{y'}(\theta) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix}. \tag{4.22}$$

The final step is a rotation by the roll angle ϕ around the new X"-axis (the axis after the second rotation). This rotation rolls the satellite about its forward direction, adjusting its tilt left or right. The result of this third rotation is the final alignment of the satellite's body frame with the LVLH frame, i.e;

$$R_{x''}(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \tag{4.23}$$

At this point we can express the angular velocity of the body frame with respect to the LVLH. We know that, in general, the angular velocity of a reference frame can be obtained as

$$\boldsymbol{\omega_b} = \dot{R}_{\mathrm{b}}^{LVLH} (R_{\mathrm{b}}^{LVLH})^{\top}, \tag{4.24}$$

where

$$\dot{R}_{\mathrm{b}}^{LVLH} = \dot{R}_{x''}(\phi) R_{y'}(\theta) R_z(\psi) + R_{x''}(\phi) \dot{R}_{y'}(\theta) R_z(\psi) + R_{x''}(\phi) R_{y'}(\theta) \dot{R}_z(\psi). \tag{4.25}$$

Thus, we obtain the following formula, that represents the relation between $\boldsymbol{\omega_b}$ and the evolution of the Euler angles:

$$\boldsymbol{\omega_b} = T(\phi, \theta)\dot{\boldsymbol{\eta}} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix}. \tag{4.26}$$

4.8 Attitude dynamics equations

The rotational dynamics of the satellite are governed by Euler's equations [37], which describe the rotational motion of a rigid body:

$$\begin{cases} I_x \frac{d\omega_x}{dt} + (I_z - I_y)\omega_y\omega_z = M_x \\ I_y \frac{d\omega_y}{dt} + (I_x - I_z)\omega_z\omega_x = M_y \\ I_z \frac{d\omega_z}{dt} + (I_y - I_x)\omega_x\omega_y = M_z \end{cases} \tag{4.27}$$

where I_x, I_y, I_z are the principal moments of inertia of the satellite, ω_x, ω_y, ω_z are the time derivatives of the angular velocities and M_x, M_y, M_z are the external torques acting on the satellite. These equations are essential for understanding how the body's angular momentum and torques interact. However, Euler's equations alone do not provide a complete solution for determining the body's orientation with respect to the reference frame. While angular velocity describes how fast and in what direction a body is rotating, it does not tell us how the body is oriented at a particular moment. This information can be found by computing the Euler angles. From equation (4.26), we can calculate the derivative of the Euler angles:

$$\dot{\boldsymbol{\eta}} = (T(\phi, \theta))^{-1}\boldsymbol{\omega_b} \tag{4.28}$$

Solving the differential equation in (4.28) we obtain the vector of angles $\boldsymbol{\eta} = [\phi, \theta, \psi]$. In summary, the system of differential equations that govern the attitude dynamics is:

$$\begin{cases} \dot{\boldsymbol{\eta}} = (T(\phi, \theta))^{-1}\boldsymbol{\omega_b} \\ \dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}(\mathbf{M} - \boldsymbol{\omega} \times (\mathbf{I}\boldsymbol{\omega})) \end{cases} \tag{4.29}$$

Chapter 5

Numerical Results

5.1 Testing set up

In this section, the validation of the UKF (illustrated in Section 3.2) against the one provided by the Matlab Control System Toolbox is presented. The filter and smoother have been applied to a double integrator system

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) + q_1(t) \\ \dot{x}_2(t) &= \frac{F}{m} + q_2(t) \\ y(t) &= x_1(t) + r(t), \end{aligned} \tag{5.1}$$

transformed in a discrete-time system and numerically integrated through Euler explicit method with a $\delta t = 0.1$s sampling time. x_1 represents the position and x_2 the linear velocity. F=1N is the constant force applied on the mass m=10kg of the considered system. The initial conditions of the system are $\mathbf{x}_0 = [x_1^0, x_2^0]^\top = [0,0]^\top$. The measurement function linearly associates the position state x_1 to the observed variable y. To reproduce sensor digital sampling, observations are made each time interval $\Delta t = 0.1$s. Both the state $\mathbf{x} = [x_1, x_2]^\top$ and the output y are affected by noise. The true process noise components of the vector $\mathbf{q} \sim \mathcal{N}(\mathbf{0}_{2\times 1}, \Sigma_q)$, with $\Sigma_q \in \mathbb{R}^{2\times 2}$, $\Sigma_q \succ 0$, enter the equation linearly as additive terms. The variable q_1 and q_2 represent respectively the position and velocity random noise input with standard deviations of $\sigma_{q_1} = 0.32$m and $\sigma_{q_2} = 0.32$m/s. The true measurement noise r is a scalar probabilistic variable with 0 mean and $\sigma_r = 1$m standard deviation. It is added to the output variable y.

5.2 Filter validation and analysis

The UKF algorithm implementation is validated through the corresponding Matlab Control Toolbox. The state value starting estimates are $\hat{\mathbf{x}}_{0|-1} = [0,0]^\top$. The notation $0|-1$ is used to describe the first predicted variable. The values of $\alpha = 10^{-3}$, $\beta = 2$ and $\kappa = 0$ (see section 3.2) parameters are the optimal ones in case of a Gaussian distribution. The modeled measurement and process noise covariance matrices, respectively R and Q, are chosen for simplicity constant and equal to the true noise dispersion matrices (σ_r and Σ_q). The filter reliability is confronted in Figure 5.1 through the time behaviour of the filter states quadratic deviation with respect to the Control toolbox estimates.

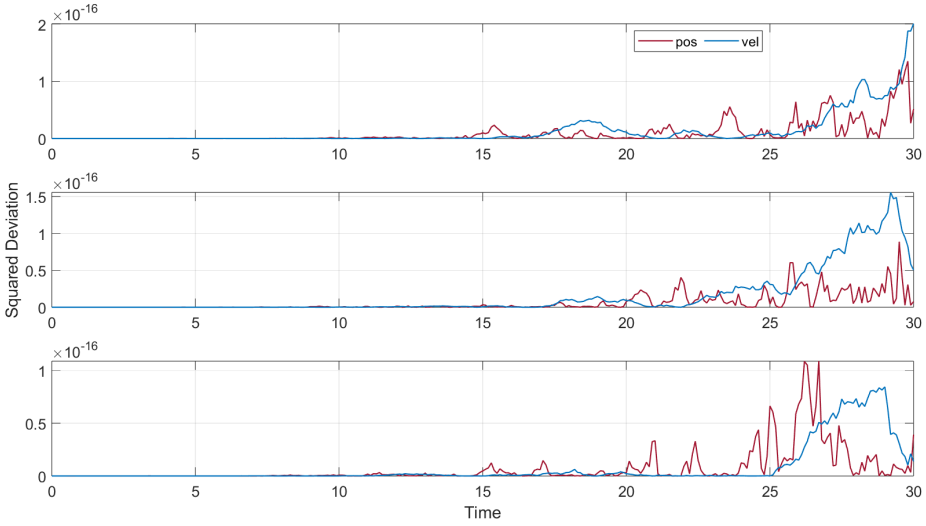

Figure 5.1: Validation filter against control toolbox algorithm accuracy for different values of the estimated standard deviation $\hat{\sigma}_{0|-1} \in [0.001,1,3.16]$ In the upper figure $\hat{\sigma}_{0|-1}$=0.001, in the middle one $\hat{\sigma}_{0|-1}$=1 and in the lower one $\hat{\sigma}_{0|-1}$=3.16.

The graphs analyze the deviation behaviour for three different initialization values of $\hat{\sigma}_{0|-1}$ (namely 0.001, 1, 3.16 for each state). As it can be seen by the squared deviation values both for position and velocity, the two algorithms lead to the same result. The Table 5.1 shows the state root means square error (RMSE) increase with the growth of the standard deviation initialization value. In Figure 5.2 the filter accuracy is tested with the squared deviation of the estimate states $\hat{\mathbf{x}}_{k|k} = [\hat{x}^1_{k|k}, \hat{x}^2_{k|k}]^\top$ with respect to the true vector $\mathbf{x}_k = [x^1_k, x^2_k]^\top$. For smaller values of the predicted covariance matrix initialization $\hat{\mathbf{P}}_{0|-1}$, the squared deviation also reduces. In particular, for $\hat{\sigma}_{0|-1} = 0.001$ the peaks are 0.9 m (position) and $0.25\,m/s$ (velocity). The values are the same for $\hat{\sigma}_{0|-1} = 1$. For

RMSE between filter and Control Toolbox estimates			
	$\hat{\sigma}_{0\|-1} = 0.001$	$\hat{\sigma}_{0\|-1} = 1$	$\hat{\sigma}_{0\|-1} = 3.16$
$RMSE_{x_1}$	$2.2 \cdot 10^{-9}\, m$	$2.3 \cdot 10^{-9}\, m$	$2.7 \cdot 10^{-9}\, m$
$RMSE_{x_2}$	$1.7 \cdot 10^{-9}\, m/s$	$1.8 \cdot 10^{-9}\, m/s$	$2.9 \cdot 10^{-9}\, m/s$

Table 5.1: Validation of UKF filter through RMSE with respect to Control toolbox algorithm.

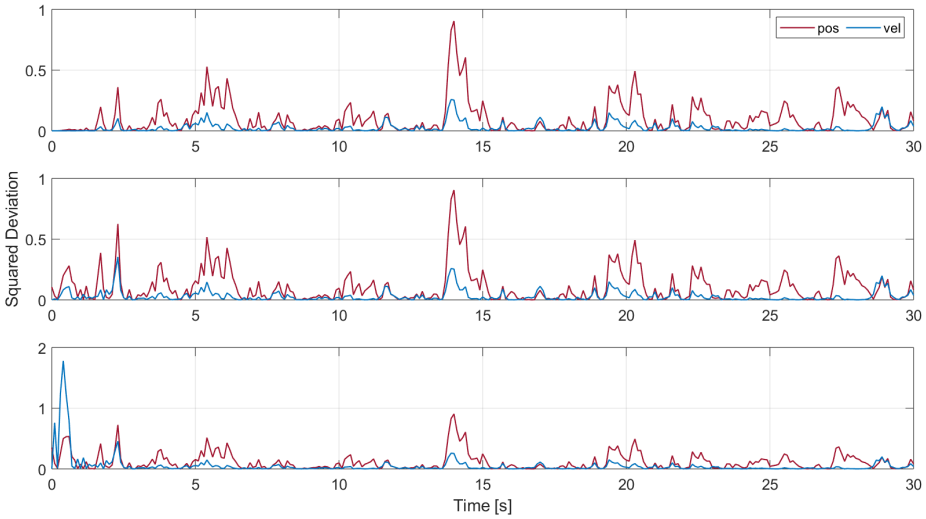

Figure 5.2: Filter accuracy with respect to the true values for different values of the estimated standard deviation $\hat{\sigma}^2_{0|-1} \in [0.001,1,3.16]$ In the upper figure $\hat{\sigma}_{0|-1}$=0.001, in the middle one $\hat{\sigma}_{0|-1}$=1 and in the lower one $\hat{\sigma}_{0|-1}$=3.16.

$\hat{\sigma}_{0|-1} = 3.16$ given the completely wrong initial estimation, the biggest values are 1.78 m for the velocity while still being $0.9\, m/s$ for the position. As for Table 5.1, also 5.2 shows the RMSE values increase related to the standard deviation initialization. The Figure 5.3 for the double integrator system, shows the time behavior of the filtered state. It is highlighted the filtered standard deviation and expected value trend. The filtered position is closer to the real state than the measurements are. The filtered velocity behavior is also similar to the real one. After a transient of $\Delta T = 5.8s$, the value of the filtered state standard deviations stabilize to $\sigma_1 = 0.4$m and $\sigma_2 = 0.42$m/s, which corresponds to a 60% decrease with respect to the measurement dispersion.

RMSE between filter estimation and true values			
	$\hat{\sigma}_{0\|-1} = 0.001$	$\hat{\sigma}_{0\|-1} = 1$	$\hat{\sigma}_{0\|-1} = 3.16$
$RMSE_{x_1}$	$0.3366\,m$	$0.3494\,m$	$0.3574\,m$
$RMSE_{x_2}$	$0.1866\,m/s$	$0.2026\,m/s$	$0.2508\,m/s$

Table 5.2: Filter RMSE for different predicted standard deviation initialization ($\hat{\sigma}_{0|-1}$).

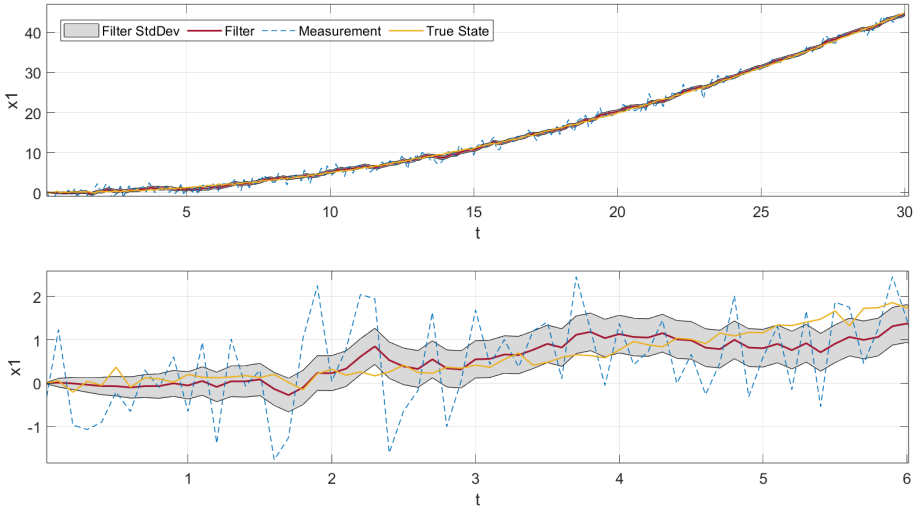

Figure 5.3: Filtered position behaviour and zoom in $t \in [0,6]$.

5.3 Smoother and filter comparison

Similarly to what was done for the filter, the behaviour of the smoother is analyzed observing the predictions with the associated standard deviations (Figure 5.4). The state trend is shown for a number of intervals $n_{cycles} = 3$. The initialization values of the smoother are for the covariance $\mathbf{P}_0^s$ entries 10^{-6} and for the mean values $\mathbf{x}_0^s = [0,0]^\top$. After the same transient for which the filtered standard deviation is calculated ($\Delta T = 5.8\,s$), the mean values of σ_1^s and σ_2^s are respectfully 0.27 m and $0.32\,m/s$, indicating a 33% and 24% improvement with respect to the filtered state standard deviations. Figure 5.5 shows how, the smoother standard deviation strip is contained in the filtered one. It follows that the UKS provides greater estimate confidence than the UKF. The amount of intervals imply a discontinuity behaviour of the strip, where it widens as soon as it shrinks. The reason lies on the initialization of the crossover covariance matrix. Since the matrix is initialized

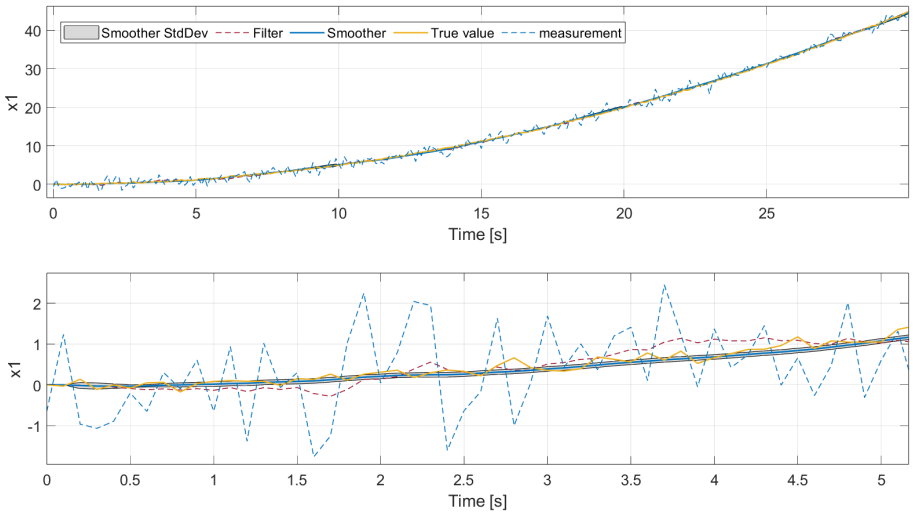

Figure 5.4: Smoothed position behaviour and zoom in $t \in [0,6]$.

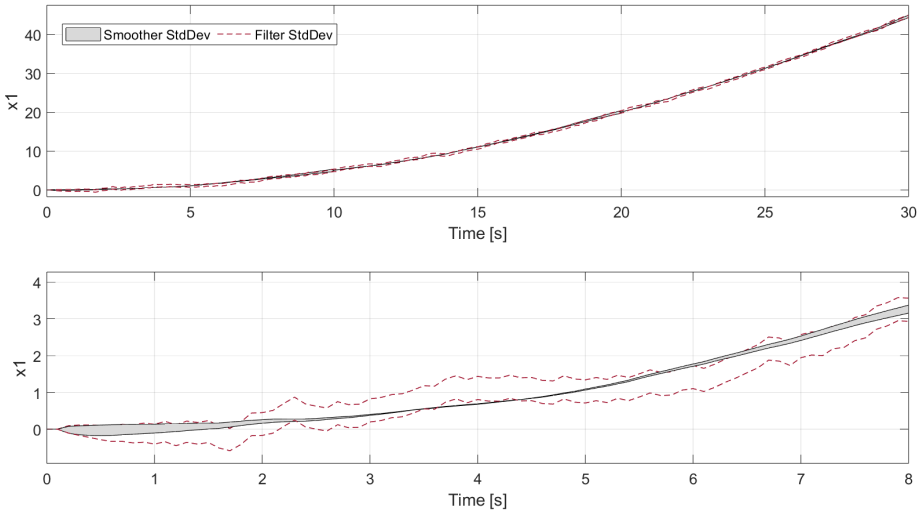

Figure 5.5: Smoother and Filter standard deviation comparison in $t \in [0,8]$.

for every interval, it might not be the same to the one corresponding to the last time step of the previous interval. The RMSE bar graph in Figure 5.6 compares the accuracy of the filter and smoother as the number of intervals, n_{cycles}, varies. Notably, as the number of intervals increases, the error also increases, eventually converging to the same values observed for the filter.

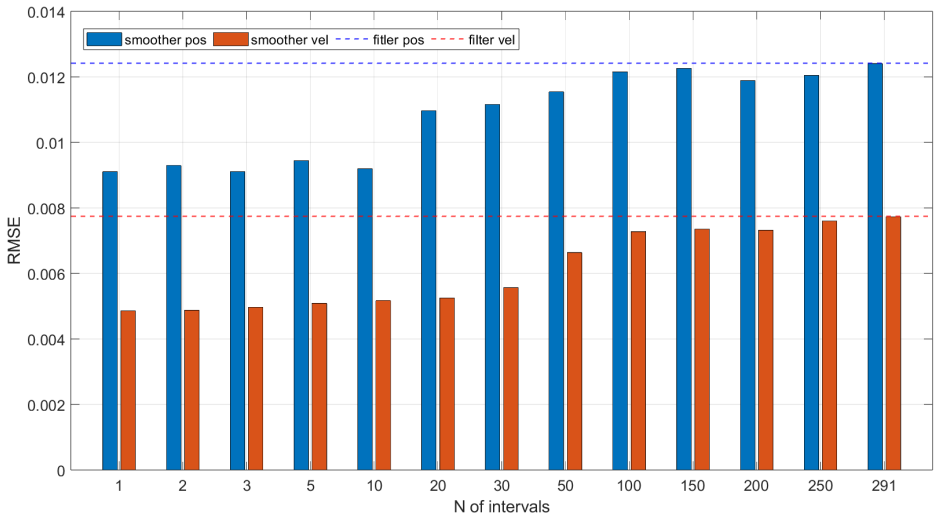

Figure 5.6: As the size of each interval is equal to one time instant, the filter and the smoother RMSE are indistinguishable.

The size and thus the number of the time intervals over which the algorithm smooths out the function is chosen with respect to the specification. As the smoothing interval grows, more information from future observations becomes available to refine the state estimates at earlier time steps. Specifically, the UKS adjusts its estimates by considering how the system evolved and reacted to control inputs over the entire interval. However, the main trade-off with increasing the smoothing interval is that it inherently introduces more lag in the estimation process. It can be understood as the delay between when a state occurs and when a refined estimate for that state can be generated.

Total time elapsed for different amount of data		
N	ΔT_{UKF}	ΔT_{UKS}
601	0.36 s	0.39 s
6×10^3	2.2 s	2.37 s
6×10^4	18.32 s	19.4 s
6×10^5	181.76 s	192.23 s
6×10^6	1901.36 s	2050.05 s

Table 5.3: Comparison between UKF and UKS time complexity for different number of data.

The code is compiled for different amounts of data (Table 5.3). Both filter and smoother elapsed time grows one order of magnitude as the data size increases the same way. The smoother elapsed time in each case is 5% greater than the filter. The wider is the size, the better is the smoothness action, but the greater is the lag with respect to the current time step. The computational weight of the new algorithm is then heavier than the previous code. A brief discussion over the application feasibility is needed. Figure 5.7 graph shows how the smoother elapsed time decreases as the number of time intervals increase, until it becomes indistinguishable from the filter.

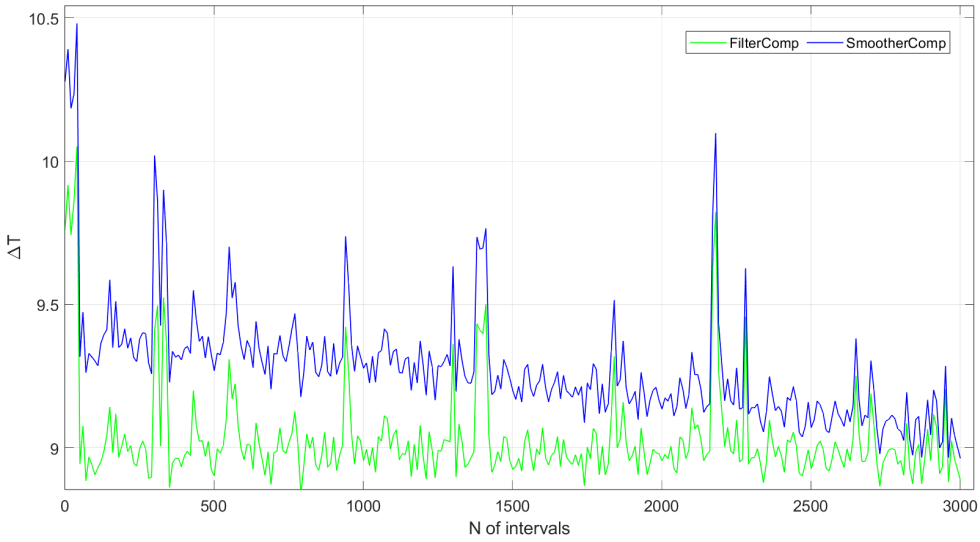

Figure 5.7: The graph shows how the algorithm elapsed time decreases as the number of intervals increases.

The reason is that for narrower intervals the amount of data inside the nested loops reduces.

5.4 Huber-based UKF and UKS testing set up

In the following sections both the Huber-based UKF and UKS are compared to the classical UKF and UKS algorithms in various scenarios with respect to the sensor noise $r(t)$ distribution defined in Section 4.1. The confrontation has been carried out on the same double integrator system illustrated in equation (5.1). In this case, the sensor noise at each time instant t is sampled from a contaminated probability distribution composed by two Gaussian distributions. The first one has the same characteristics of the one used to sample the noise $r(t)$ in (5.1), namely it

has $\mu_1 = 0$ mean and $\sigma_{r1} = 1m$ standard deviation. The second one has $\mu_2 = 0$ mean but an higher standard deviation of $\sigma_{r2} = 10m$. The following equation represents a system where each value of the standard deviation is associated with the probability that a sensor noise is generated according to that distribution.

$$\begin{cases} \sigma_r = 1\,m & w.p. \quad p \\ \sigma_r = 10\,m & w.p. \quad 1-p \end{cases} . \tag{5.2}$$

where p and $1-p$ represent the probability that the sensor noise at time t follows respectively the first or the second distribution [32]. The assumed noise standard deviation is fixed at $\sigma_R = 1m$.

5.5 Influence of γ in the HUKF estimation

This section will investigate the influence of the threshold parameter γ on the HUKF estimate. Figure 5.8 shows the behaviour of the first variable of the double integrator system for $\gamma = 0.1$ and $p = 1$. The distribution is Gaussian and it follows the filter assumed variance (R=1)

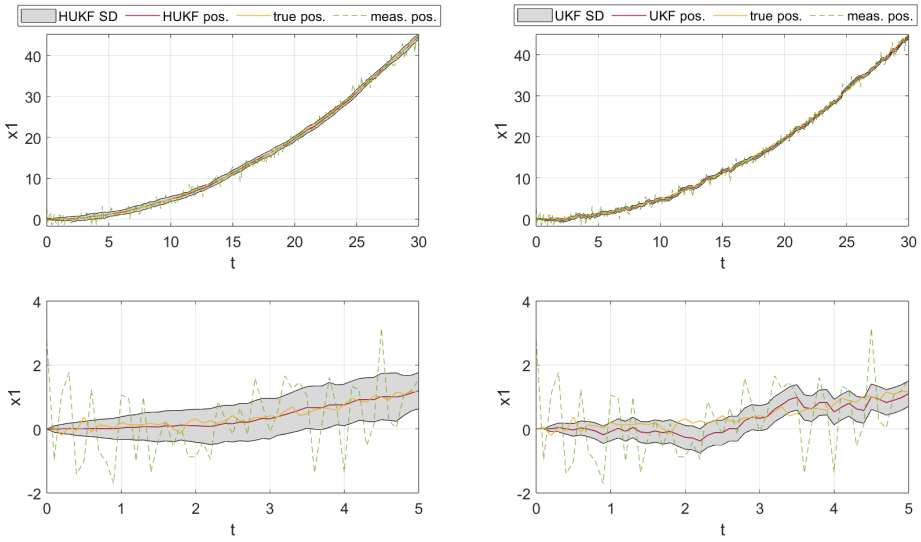

Figure 5.8: Confrontation of position behaviour between Huber-based UKF and classical UKF for $\gamma = 0.1$ and $p = 1$.

As underlined in the final section of chapter 4, γ influences the estimate standard deviation and robustness against uncertainties. In a Huber-based Unscented Kalman Filter, it defines the point where the loss function switches from quadratic to linear, treating the measurement as a possible outlier. Since γ is very low, the

loss function transitions more quickly to the linear part, treating even small errors as potential outliers. This results in reduced sensitivity to smaller errors, thereby increasing the estimator's variance. Instead, Figure 5.9 captures the estimate in the case of $\gamma = 1.345$.

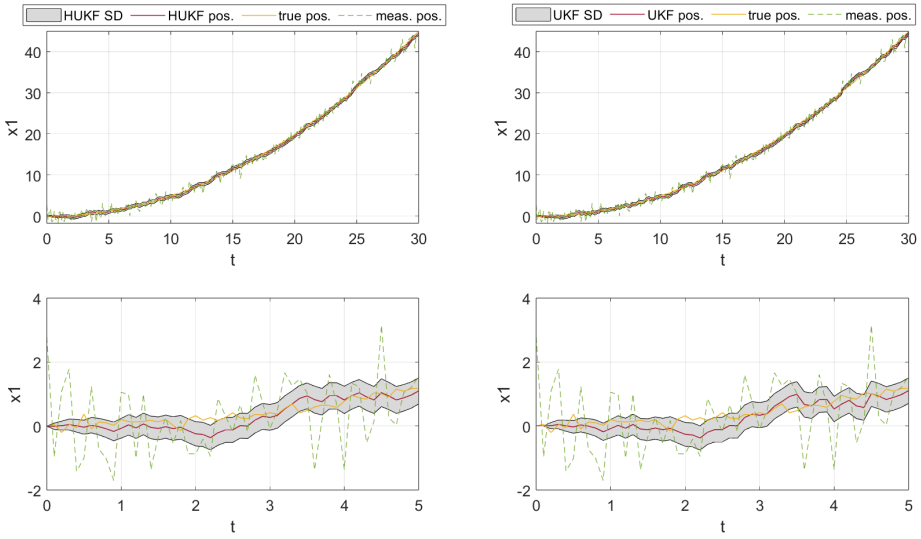

Figure 5.9: Confrontation of position behaviour between Huber-based UKF and classical UKF for $\gamma = 1.345$ and $p = 1$.

As the value is close to the assumed variance, most errors fall within the quadratic region. This makes the filter behave more like a traditional Kalman Filter, which assumes Gaussian-distributed noise and minimizes the estimator's variance. The estimator responds optimally to the noise distribution, effectively correcting errors and maintaining a low variance. Figure 5.8 and 5.9 show how it is important to choose the correct value of the threshold γ. When it reduces too much, the variance increases exponentially. When the value is too high, the accuracy reaches the level of the classical UKF.

5.6 Influence of p in HUKF

The confrontation is carried out for different values of the probability p defined in equation (3.28) to evaluate their behaviour as it changes. In particular, three values of $p = [0,0.5,0.9]$ have been chosen. Table 5.4 and 5.5 contain the values of the RMSE and standard deviation for both UKF and HUKF. For a value of **p=0.9**, the noises are sampled with a probability of $p = 0.1$ from the Gaussian distribution with larger variance.

RMSE and σ mean of HUKF for $p = [0,0.5,0.9]$ and $\gamma = 1.345$				
	$RMSE_{pos}$	$RMSE_{vel}$	σ_{pos}	σ_{vel}
$p = 0.9$	$0.4465\,m$	$0.2206\,m/s$	$0.4054\,m$	$0.4104\,m/s$
$p = 0.5$	$0.7785\,m$	$0.3604\,m/s$	$0.4636\,m$	$0.4244\,m/s$
$p = 0$	$2.2849\,m$	$0.8292\,m/s$	$0.5812\,m$	$0.45\,m/s$

Table 5.4: Comparison of HUKF accuracy and uncertainty for different values of p.

RMSE and σ of UKF for $p = [0,0.5,0.9]$ and $\gamma = 1.345$				
	$RMSE_{pos}$	$RMSE_{vel}$	σ_{pos}	σ_{vel}
$p = 0.9$	$1.235\,m$	$0.6371\,m/s$	$0.3894\,m$	$0.4065\,m/s$
$p = 0.5$	$2.5681\,m$	$1.2416\,m/s$	$0.3894\,m$	$0.4065\,m/s$
$p = 0$	$3.556\,m$	$1.7033\,m/s$	$0.3894\,m$	$0.4065\,m/s$

Table 5.5: Comparison of UKF accuracy and uncertainty for different values of p.

The Huber-based filter reduces the RMSE by 63.8% for position and by 65.3% for velocity with respect to the classical filter (from the RMSE values for the HUKF and UKF with $p = 0.9$ in Table 5.4 and 5.5). The standard deviation grows by the 4.1% for the position and by the 1% for the velocity with respect to the UKF (from the standard deviation values for the HUKF and UKF with $p = 0.9$ in Table 5.4 and 5.4). A value of **p=0.5** indicates that the noises are sampled with the same probability both from the Gaussian distribution with $\sigma_r = 10$ and from the one with $\sigma_r = 1$. It represents the case of perfect Gaussian mixture. The Huber-based algorithm maintains a good accuracy with an RMSE of 0.7785 m and $0.3604\,m/s$ while the the classical UKF behaviour worsens (RMSE of 2.5681 m and $1.2416\,m/s$). This is due to the increase, across different time instants, in the number of uncertain measurements that skew the expected value of the estimator. The case for **p=0** is the worst scenario of all. The distribution is Gaussian, however its characteristics follows the p.d.f with the greatest variance. The measurement noise sample is not produced from the assumed distribution with variance $R = 1$. The accuracy degradation of the robust approach is evident. The Huber-based filter RMSE is $2.2849\,m$ for the position and $0.8292\,m/s$ for the velocity. Still, it is a better performance than that of the classical UKF with an RMSE of $3.556\,m$ for the first state and $1.7033\,m/s$ for the second one. Figure 5.10 shows a confrontation between the two algorithm for the three values of p. It can be seen how the UKF estimate deviates from the true values as the value of p increases. This performance worsening is much less noticeable in the HUKF

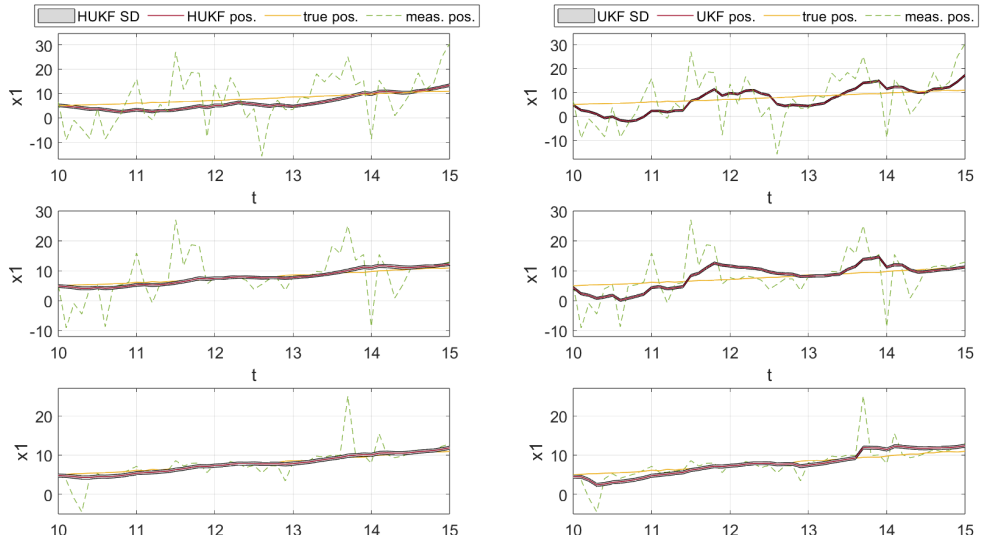

Figure 5.10: HUKF and UKF confrontation for $\gamma = 1.345$ and increasing values of $p = [0,\ 0.5,\ 0.9]$ from top to bottom.

graphs (at least for $p = [0.5,\ 0.9]$). In particular, the classical UKF exhibits lower accuracy compared to the Huber-based filter due to its lack of robustness in the presence of uncertain measurements. In contrast, the HUKF enhances robustness by adjusting the weight ψ_i of each measurement based on the magnitude of its residual. This adaptive weighting allows the HUKF to mitigate the impact of outliers and noise, resulting in a more accurate and reliable estimation process with respect to the UKF. This means that a robust approach not only preserves accuracy against probable uncertain measurements, but also against a bad modeling of the measurement noise variance. When it comes to standard deviation,while it remains constant in the classical UKF, that of the HUKF increase. This increase reflects the highlighted robustness of the HUKF against outliers. The filter adapts by inflating the standard deviation to account for the uncertainty, effectively reducing the influence of outliers on the final estimation. The reason is that the HUKF has the ability to adapt the Kalman gain $\tilde{K}$ with respect to the residual magnitude changing the output covariance matrix. The UKF has no mathematical tools to do so.

5.7 Influence of p in HUKS

When it comes to the Huber-based UKS, the behavior with respect to changes in the values of p is similar to that of the HUKF. However, the standard deviation in the Huber-based UKS is smaller compared to the corresponding HUKF, indicating

a more refined and accurate estimation process.

RMSE and σ of HUKS for $p = [0,0.5,0.9,1]$ and $\gamma = 1.345$				
	$RMSE_{pos}$	$RMSE_{vel}$	σ_{pos}	σ_{vel}
$p = 1$	$0.2946\,m$	$0.1842\,m/s$	$0.2748\,m$	$0.3158\,m/s$
$p = 0.9$	$0.3377\,m$	$0.2083\,m/s$	$0.2822\,m$	$0.3181\,m/s$
$p = 0.5$	$0.6461\,m$	$0.3967\,m/s$	$0.3252\,m$	$0.3324\,m/s$
$p = 0$	$1.9611\,m$	$0.9223\,m/s$	$0.4103\,m$	$0.3599\,m/s$

Table 5.6: Comparison of HUKS accuracy and uncertainty for different values of p.

RMSE and σ of UKS for $p = [0,0.5,0.9,1]$ and $\gamma = 1.345$				
	$RMSE_{pos}$	$RMSE_{vel}$	σ_{pos}	σ_{vel}
$p = 1$	$0.2992\,m$	$0.1842\,m/s$	$0.271\,m$	$0.3144\,m/s$
$p = 0.9$	$0.9126\,m$	$0.6\,m$	$0.271\,m$	$0.3144\,m/s$
$p = 0.5$	$2.0526\,m$	$1.3489\,m/s$	$0.271\,m$	$0.3144\,m/s$
$p = 0$	$2.8038\,m$	$1.5852\,m/s$	$0.271\,m$	$0.3144\,m/s$

Table 5.7: Comparison of UKS accuracy and uncertainty for different values of p.

The smoother objective is to smooth the trajectory and to reduce the estimator uncertainty and thus its variance. The last task is of great importance as the robust approach tends to increase the estimator variance in order to reduce the influence of possible outliers. As it can be seen by the confrontation between classical UKS and Huber-based UKS in Figure 5.11 the best smoothing action is provided by the latter. The reason lies on the inability of the classical UKS to intercept and penalize uncertain observations. It creates a too noisy behaviour to be smoothed out. As shown in Table 5.6, the HUKS achieves an RMSE for a p value of 0 of 1.961 m for position and $0.9223\,m/s$ for velocity. In contrast, Table 5.7 shows that the UKS RMSE values are higher, at 2.8038 m for position and $1.5852\,m/s$ for velocity. As for the standard deviation, both the HUKS and the UKS with respect to HUKF and UKF, experience a reduction of 30% for the first system state and 20% for the second one.

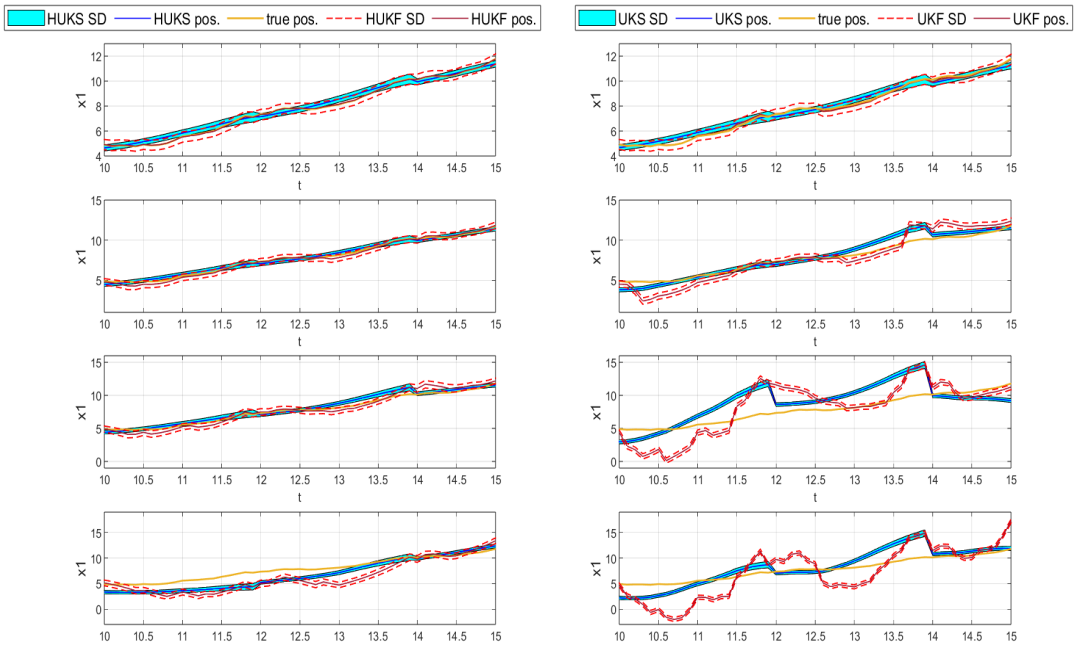

Figure 5.11: HUKS and UKS confrontation for $\gamma = 1.345$ and decreasing values of $p = [0,\ 0.5,\ 0.9,\ 1]$ from top to bottom.

5.8 Complexity increase due to Huber-based algorithms

In terms of computational complexity the Huber-based UKF adds time to complete a single iteration. The first is needed to compute the single weight ψ_i and the second to evaluate the all the component of the matrix Ψ. Finally the matrix Ψ has to be inverted to obtain the updated value of the measurement noise covariance matrix $\tilde{R}$.

Elapsed time for single iteration (s)				
	ΔT_{UKF}	ΔT_{HUKF}	ΔT_{UKS}	ΔT_{HUKS}
$p = 1$	0.00262 s	0.00226 s	0.00243 s	0.00255 s
$p = 0.9$	0.00222 s	0.00231 s	0.00238 s	0.0026 s
$p = 0.5$	0.00225 s	0.0023 s	0.00241 s	0.00258 s
$p = 0$	0.00234 s	0.00234 s	0.0025 s	0.00264 s

Table 5.8: Comparison of the algorithms time complexity for different values of p.

The Table 5.8 shows that in percentage the difference between the two elapsed

times does not surpass the 4% in all four cases. When it comes to the HUKS, the added complexity lies on the computation of the process noise covariance matrix $\tilde{Q}$. When comparing the elapsed time per iteration between the HUKF and HUKS for all the values of p, the difference in execution time does not exceed 9%.

Chapter 6

Satellite Dynamical Results

6.1 Testing set up

In this section, the results of the filter and smother are presented with respect to a physical system that incorporates both Newton (4.19 in Chapter 4) and Euler equations (4.27 also in 4) for the estimation of the orbiting satellite pose. The system is n-dimensional with $n = 12$, non-linear and time-invariant. It can be written as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \ddot{\mathbf{r}} \\ \dot{\boldsymbol{\eta}} \\ \dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} + \mathbf{q}_v \\ -\frac{\mu}{|\mathbf{r}|^3}(\mathbf{r} + \mathbf{q}_r) + \mathbf{F}_{\text{ext}} \\ \mathbf{T}(\boldsymbol{\eta} + \mathbf{q}_\eta)^\top (\boldsymbol{\omega} + \mathbf{q}_\omega) \\ \mathbf{I}^{-1}\left(\mathbf{M}_{\text{ext}} - (\boldsymbol{\omega} + \mathbf{q}_\omega) \times (\mathbf{I}(\boldsymbol{\omega} + \mathbf{q}_\omega))\right) \end{bmatrix} \tag{6.1}$$

where $\mathbf{r} = [x,\, y,\, z]^\top$ represents the position state vector, $\boldsymbol{\eta} = [\phi,\, \theta,\, \psi]^\top$ contains the Euler angles and $\boldsymbol{\omega} = [\omega_x,\, \omega_y,\, \omega_z]^\top$ the angular velocity of the body frame with respect to the LVLH reference frame defined in Chapter 4. $\mathbf{I}$ defines the inertia matrix and its diagonal entries I_x, I_y, I_z are the the principal moments of inertia about the body's x-, y-, and z-axes, respectively. The rotation matrix $\mathbf{T}(\boldsymbol{\eta})$ is a transformation matrix that relates the rates of change of the Euler angles $\dot{\boldsymbol{\eta}}$ to $\boldsymbol{\omega}$. The vector, of dimension n, $\mathbf{q} = [\mathbf{q}_r,\, \mathbf{q}_v,\, \mathbf{q}_\eta,\, \mathbf{q}_\omega]$ define the process noise of position, velocity, orientation and angular velocity. The mean value is $\mathbf{0}_{nxn}$ and the covariance matrix $\boldsymbol{\Sigma}_q$ elements are

$$\boldsymbol{\Sigma}_q = \begin{bmatrix} 25\mathbb{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & 25\mathbb{I}_3 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & 5\times10^{-6}\mathbb{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & 5\times10^{-6}\mathbb{I}_3 \end{bmatrix}. \tag{6.2}$$

The measured output y is expressed instead as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_r + \mathbf{n}_r \\ \mathbf{y}_\eta + \mathbf{n}_\eta \end{bmatrix}, \tag{6.3}$$

where $\mathbf{n} = [\mathbf{n}_r, \mathbf{n}_\eta]$ is a vector of dimension m with mean value of $\mathbf{0}_{mxm}$ and covariance matrix $\mathbf{\Sigma}_n$ with elements

$$\mathbf{\Sigma}_n = \begin{bmatrix} \sigma_{n_r}^2 \mathbb{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \sigma_{n_\eta}^2 \mathbb{I}_3 \end{bmatrix}, \tag{6.4}$$

where σ_{n_r} and σ_{n_η} identify the standard deviation of the sensor giving respectively position and orientation measures. The sensor noise follows a Gaussian mixture distribution. Thus σ_{n_r} , similarly to what is shown in (5.2), is expressed as:

$$\begin{cases} \sigma_{n_r} = 10\,m & w.p. \quad p \\ \sigma_{n_r} = 100\,m & w.p. \quad 1-p \end{cases} \tag{6.5}$$

while σ_{n_η} is represented as

$$\begin{cases} \sigma_{n_\eta} = 0.008\,rad & w.p. \quad p \\ \sigma_{n_\eta} = 0.025\,rad & w.p. \quad 1-p \end{cases} \tag{6.6}$$

The initial condition values for the system is shown in Table 6.1. The filter counterparts are defined in Table 6.2. When it comes to the position, the initial coordinates is given by transforming the variable from the PQW system (assumed to be $\mathbf{r}_{\text{PQW}}^{in}$ expressed in equation (4.11) in chapter 3) to the ECI coordinate frame through the Keplerian elements described in chapter 3. The initial $\boldsymbol{\omega}$ is assumed to be $[0,\, -\omega_0,\, 0]^\top$ where $\omega_0 = 0.0011$ is the orbital angular velocity of the satellite.

System initial conditions	
Variable	**Value**
$\mathbf{r}_{\text{ECI}}^{in}$	$[-6.46\times10^6,\, 2.896\times10^6,\, 0]^\top m$
$\mathbf{v}_{\text{ECI}}^{in}$	$[0,\, 0,\, 0]^\top m/s$
$\boldsymbol{\eta}^{in}$	$[0,\, 0,\, 0]^\top rad$
$\boldsymbol{\omega}^{in}$	$[0,\, -0.0011,\, 0]^\top rad/s$

Table 6.1: Variable states initial conditions.

Filters Initial conditions	
Variable	**Value**
$\mathbf{r}_{0\|0}$	$\mathbf{y}_{pos}^{in}$ (in. position measured output)
$\mathbf{v}_{0\|0}$	$[0,\ 0,\ 0]^\top m/s$
$\boldsymbol{\eta}_{0\|0}$	$\mathbf{y}_{or}^{in}$ (in. orientation measured output)
$\boldsymbol{\omega}_{0\|0}$	$[0,\ 0,\ 0]^\top rad/s$
$\sigma_{0\|0}^{lin}$	$100m$ and $100m/s$
$\sigma_{0\|0}^{ang}$	$10^{-6}rad$ and $10^{-6}rad/s$

Table 6.2: Initial estimated mean values and uncertainties of the filters for both position and orientation.

6.2 Filter comparison

Let us analyze the UKF and HUKF algorithm in the worst scenario of non Gaussian noise which is p=0. In this case the sensor noise has standard deviation of $\sigma_{n_r} = 100\,m$ and $\sigma_{n_\eta} = 0.025\,rad$ respectively for position and orientation. The results presented in Figures 6.1 and 6.2 show a clear improvement in the state estimation performance of the Huber-UKF compared to the standard UKF.

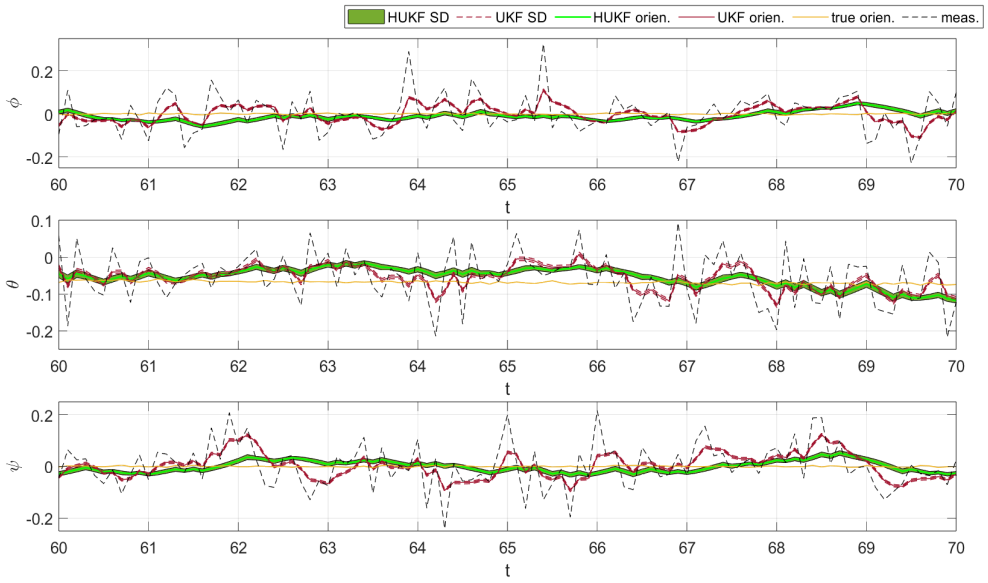

Figure 6.1: HUKF and UKF confrontation for orientation in time interval $t = [60,70]$.

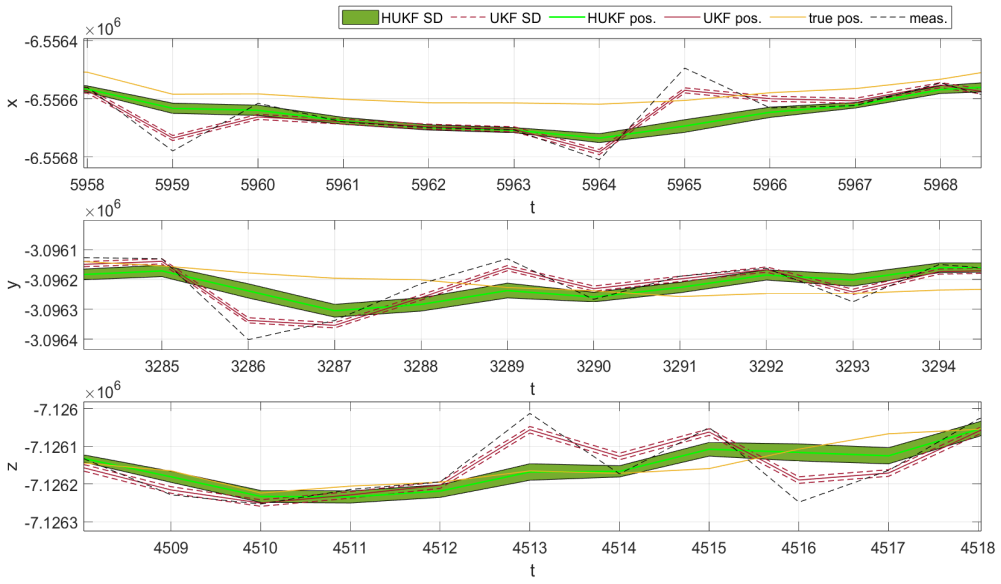

Figure 6.2: HUKF and UKF confrontation for position at different time interval $\Delta t = 10$.

Notably, the Huber-UKF improves estimates accuracy of 34% for the orientation problem and almost 30% for the orbit position determination as it can be seen from Table 6.3 and 6.4. The robust nature of the Huber loss function allows the filter to limit the impact of these erroneous measurements, preventing the estimation drift observed with the UKF.

UKF RMSE and standard deviation		
	RMSE	$\boldsymbol{\sigma}$
x	78.8617 m	8.6601 m
y	78.9068 m	8.6601 m
z	78.9184 m	8.6601 m
ϕ	0.0391 rad	0.0047 rad
θ	0.0388 rad	0.0047 rad
ψ	0.0373 rad	0.0047 rad

Table 6.3: UKF Accuracy and Uncertainty in an environment simulating orbital dynamics.

At the same time, this is followed by an increase of almost 40% and more than 50% respectively in the estimate uncertainty. While an increase in standard deviation is

HUKF RMSE and standard deviation		
	RMSE	$\boldsymbol{\sigma}$
x	56.2235 m	18.0843 m
y	56.0142 m	18.1605 m
z	56.3178 m	18.9848 m
ϕ	0.0258 rad	0.0077 rad
θ	0.0247 rad	0.0077 rad
ψ	0.0236 rad	0.0077 rad

Table 6.4: HUKF Accuracy and Uncertainty in an environment simulating orbital dynamics.

often viewed negatively due to the resulting decrease in confidence in the filter's estimates, it also signals the ability of the HUKF to recognize and adapt to deviations from the assumed probability distribution of sensor measurements. On the other hand, in the case of the standard UKF we need to manually set a new value for the assumed noise covariance matrices Q and R to account for the increase in data variability. In Figure 6.3 the probability density function of the sensor measuring the angle ϕ is plotted for the two cases of p=0 and p=1 together with the HUKF and UKF estimate mean standard deviation.

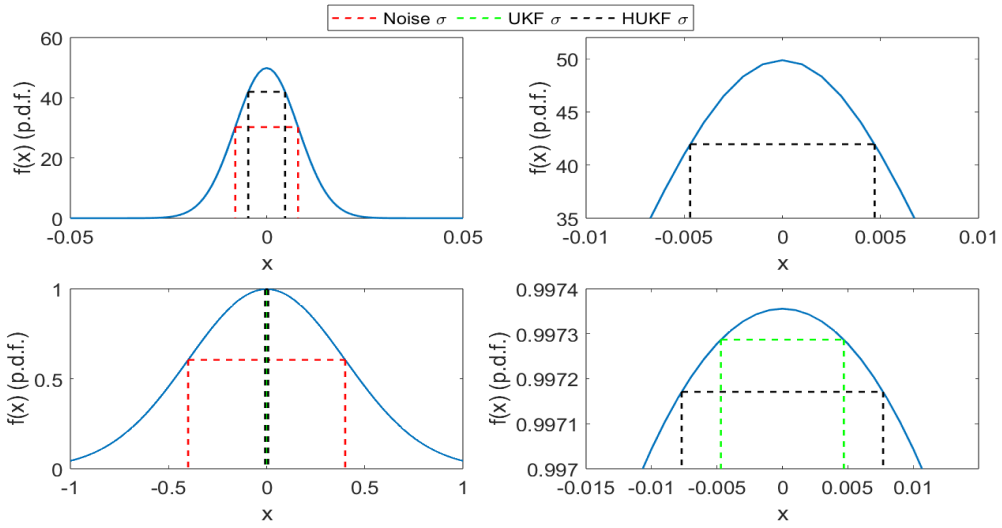

Figure 6.3: From top to bottom the sensor noise p.d.f. respectively in the p=1 and p=0 scenario. From left to right the distributions and their zoom.

As it can be seen, the two algorithms are indistinguishable in the p=1 scenario

since the sensor noise distribution follows the assumptions. However, as the data deviates from the hypothesis, the UKF still considers the assumptions valid while the HUKF does not due to its robust nature. When the HUKF deals with outliers, the Kalman gain is adjusted accordingly to reduce the impact of large residuals, but this adjustment comes with a corresponding increase in the estimated uncertainty, as the filter cannot confidently rely on the measurements anymore (as discussed in chapter 4). In case of an unreliable sensor or a space environment too harsh, the HUKF algorithm is an optimal choice if we need to safeguard the system and have an indicator on when the sensor misbehaves.

6.3 Smoother comparison

Let us consider the variable ϕ for orientation and x for position in the scenario of $p = 0$ and let us compare the HUKS algorithm with both HUKF and UKS. The Figure 6.4 shows the improvement in accuracy and precision of the HUKS over the HUKF.

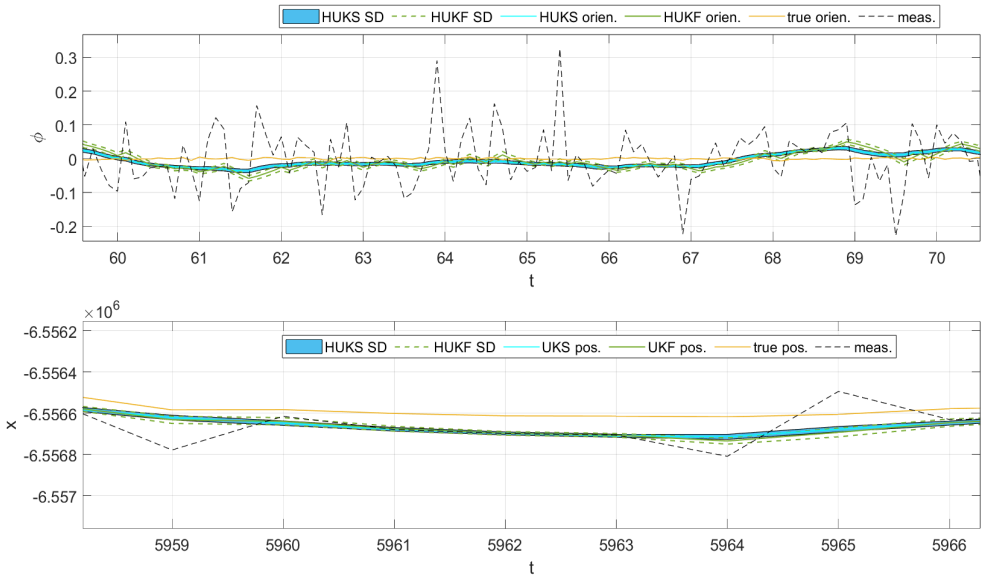

Figure 6.4: From top to bottom the comparison between HUKS and HUKF algorithm for variable ϕ and x.

This arises from the fact that the HUKS is a non-real-time algorithm. Unlike the HUKF, which processes data sequentially as it becomes available, the HUKS leverages the entire set of measurements, including both past and future data points (see chapter 4). This allows the smoother to refine the state estimates by

comparing the intermediate results from the HUKF with the expected dynamics of the system. It results in a better reduction of the high-frequency oscillations due to faulty measurements. In particular, confronting Table 6.3 with 6.4, the RMSE decreases of 20% for the variable ϕ and almost 15% for x.

HUKS RMSE and standard deviation		
	RMSE	$\boldsymbol{\sigma}$
x	48.267 m	12.7981 m
ϕ	0.0216 rad	0.005 rad

Table 6.5: HUKS Accuracy and Uncertainty in an environment simulating orbital dynamics.

Given the slow satellite dynamics, the ideal state estimation algorithm should yield smooth state transitions, accurately reflecting the gradual changes in the satellite's position and orientation. A method that introduces sharp fluctuations in the state estimate, especially in the absence of actual sudden external forces, is unrealistic for orbital systems.

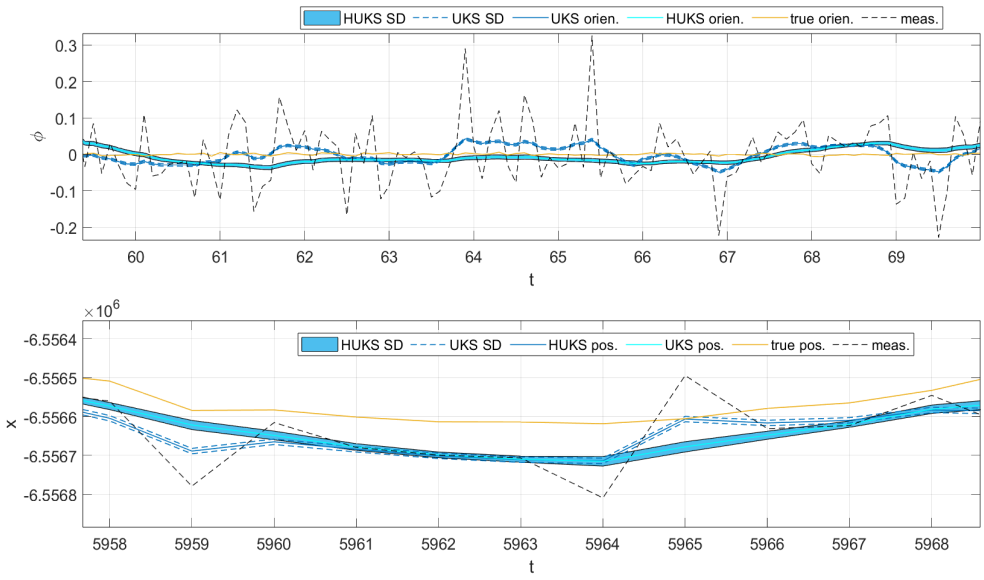

Figure 6.5: From top to bottom the comparison between HUKS and UKS algorithm for variable ϕ and x.

However, when using the HUKF in the presence of outliers in sensor data, these sudden deviations can appear due to the algorithm's real-time nature and the

influence of contaminated measurements. Therefore, the HUKS is a valid alternative in case we are interested in a more accurate representation of the dynamics. In Figure 6.5 the HUKS algorithm is compared with the UKS. Both act to smooth out the estimates of the respective real-time filter counterparts. However, the UKS deviations from the true values are still too high with respect to the HUKS. The reason is that the UKF can not distinguish from good and faulty measurements as the HUKF can. This results, for the first case, in an action where the smoothed estimates are still close to the outliers, while in the second case the mean values are closer to the true values. In conclusion, the HUKS represents an option in the case of highly unreliable measurements if we want to obtain estimates that replicate the real satellite dynamics.

6.4 Comparison of estimates with different measurement noise distributions

Let us consider all the scenarios except the one of $p = 0$ we have dealt until now. Figure 6.6 and 6.7 show the HUKF and HUKS behaviour against their respective classical filters for the Euler angle ϕ.

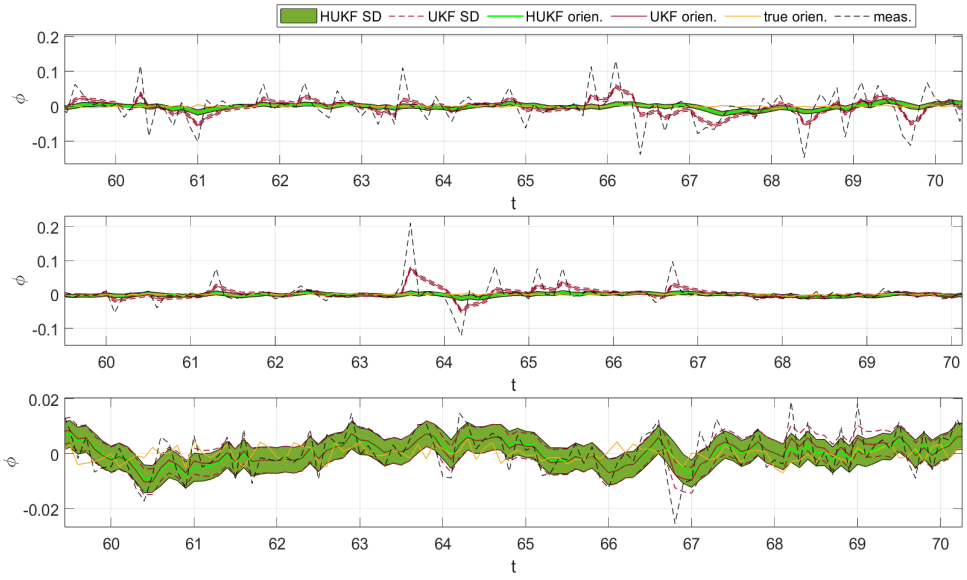

Figure 6.6: HUKF and UKF confrontation for $p = [0.5, 0.9, 1]$. From top to bottom $p = 0.5$, then $p = 0.9$ and $p = 1$.

It can be seen how for $p = 0.5$ the Huber-based filters still perform better than the UKF and UKS. However, the accuracy improvement significantly decreases for

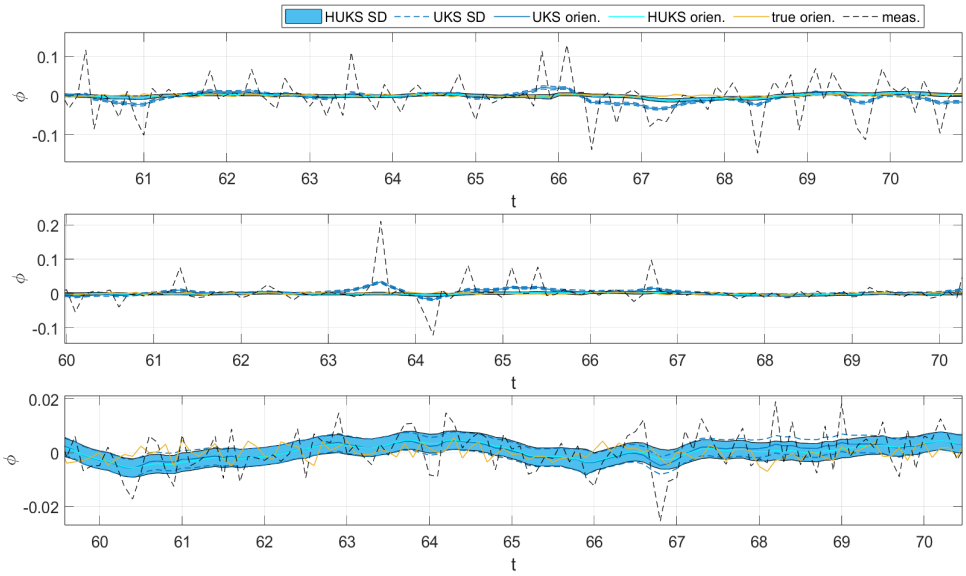

Figure 6.7: HUKS and UKS confrontation for $p = [0.5, 0.9, 1]$. From top to bottom $p = 0.5$, then $p = 0.9$ and $p = 1$.

$p = 0.9$ and for $p = 1$ they are indistinguishable one from another. Basically, in scenarios where the sensor noise variance closely aligns with the presumed model, and there is minimal deviation from the expected Gaussian noise distribution, the benefits of using HUKF or HUKS diminish. In such cases, the increased computational complexity introduced by these robust algorithms does not yield significant performance gains over the standard UKF and UKS. In particular, the additional computations include the calculations for the Huber loss function, as well as the modifications to the Kalman gain and covariance update steps that accommodate outliers. To effectively determine when the added complexity of HUKF or HUKS is unwarranted, we can propose a method that evaluates the performance-to-complexity ratio by comparing the Root Mean Square Error (RMSE) of the state estimates and the execution time of a full cycle of each algorithm. The goal is to assess how much performance gain, if any, is achieved relative to the additional computational cost. The RMSE is a standard measure of the accuracy of state estimation. It captures the difference between the estimated state and the true state over multiple time steps. The execution time is the total time it takes for a single cycle of the algorithm (either filtering or smoothing) to complete. The accuracy-to-complexty (ACR) ratio of the RMSE difference between estimates of Huber and non-Huber based algorithms and execution time difference between the same filters quantifies the benefit of HUKF/HUKS compared to UKF/UKS

$$\mathrm{ACR_{HUKF}} = \frac{\mathrm{RMSE_{UKF}} - \mathrm{RMSE_{HUKF}}}{\mathrm{T_{UKF}} - \mathrm{T_{HUKF}}} = \frac{\Delta\mathrm{RMSE}}{\Delta\mathrm{T}}. \tag{6.7}$$

$$\mathrm{ACR_{HUKS}} = \frac{\mathrm{RMSE_{UKS}} - \mathrm{RMSE_{HUKS}}}{\mathrm{T_{UKS}} - \mathrm{T_{HUKS}}} = \frac{\Delta\mathrm{RMSE}}{\Delta\mathrm{T}}. \tag{6.8}$$

Table 6.6 and 6.7 show the ACR values respectively for a HUKF and HUKS alogorithms in the scenarios of sensor noise distribution where $p = [0.5, 0.9, 1]$.

HUKF RMSE, elapsed time (T) and ACR			
p	**RMSE**	**T**	**ACR**
0.5	0.0119 rad	0.0046 s	$34.4298\,rad/s$
0.9	0.0048 rad	0.0035 s	$31.514\,rad/s$
1	0.0042 rad	0.0052 s	$0.6393\,rad/s$

Table 6.6: HUKF Accuracy and complexity comparison through parameter ACR for different p values.

HUKS RMSE, elapsed time (T) and ACR			
p	**RMSE**	**T**	**ACR**
0.5	0.0094 rad	0.0047 s	$17.3\,rad/s$
0.9	0.0038 rad	0.0035 s	$16.0792\,rad/s$
1	0.0033 rad	0.0053 s	$0.0207\,rad/s$

Table 6.7: HUKS Accuracy and complexity comparison through parameter ACR for different p values.

A low value of ACR would indicate that the added computational cost is not justified by the small improvement in accuracy, suggesting that HUKF or HUKS is not necessary for the given scenario. Conversely, a high PCR would justify the use of the more complex algorithm due to a significant improvement in accuracy. We have already said, when commenting figure , how the Huber-based algorithms performance starts decreasing for $p = 0.9$. This means that its ACR values for the HUKF and HUKS measures when the Huber-based algorithm is not useful to implement.

Chapter 7

Conclusions

7.1 The best algorithm for each scenario

In this thesis, we explored various state estimation techniques for the specific problem of pose tracking of a satellite. The analysis of the numerical results highlights the advantages of UKF, UKS, HUKF and HUKS, each suited for specific scenarios depending on the system's requirements. In particular, the radar charts in Figure 7.1 summarize the four algorithms performance in terms of accuracy, robustness and computational complexity, respectively investigating the RMSE, standard deviation and cycle elapsed time in the $p = 0$ scenario.

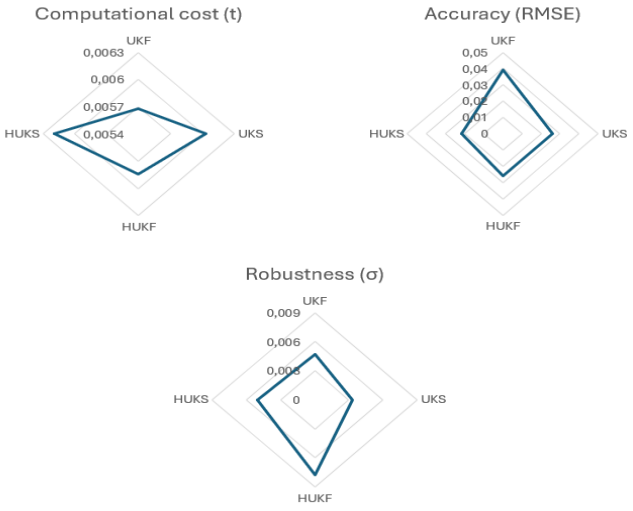

Figure 7.1: Radar graph showing the algorithms performance in terms of accuracy, robustness and computation time.

Unlike global localization problems, where the particle filter would have been a more suitable choice, the focus on pose tracking made Kalman-based methods more appropriate due to their efficiency and accuracy in handling continuous tracking of states. The Particle Filter, while powerful for non-linear problems with non-Gaussian noise, is computationally expensive and unnecessary in our case, where the satellite's initial position is relatively well-known, and tracking is the main challenge. The HUKS demonstrates superior performance when it comes to accuracy. This is captured by its ability to precisely model orbital dynamics, particularly in terms of orientation. This is critical in the final approach phase to ensure alignment between chaser and target satellites. On the other hand, HUKF shows an adequate balance between accuracy and robustness that proves to be highly effective when dealing with sensors that exhibit significant deviations from the assumed standard deviation, indicating unreliability. This makes HUKF particularly valuable in environments where sensor data may be compromised or less reliable than initially anticipated. Finally, UKF and UKS are most appropriate when the computational complexity is a crucial factor. More satellites are equipped with control units regardless of their size and good performances are demanded even when the CPU capabilities are not optimal. This filter is ideal when the trade-off between reduced RMSE and computational cost must be carefully balanced. Therefore, the choice between these filters depends on the specific requirements for accuracy, computational resources, and sensor reliability.

7.2 Further work

This thesis opens up several opportunities for further research. One promising direction for future work is the investigation of **adaptive filtering techniques**, where the filter's parameters, such as the γ threshold in the HUKF, dynamically adjust in response to changing conditions. In space missions, sensor noise characteristics or operational phases can vary significantly. For example, a satellite may encounter different levels of interference, sensor degradation, or environmental changes as it moves between mission phases, such as orbital maneuvers or docking procedures. An adaptive approach would allow the filter to automatically tune itself, enhancing its robustness and flexibility in unpredictable environments. Another crucial step for future work involves developing **real-time versions of the UKS and HUKS** optimized for onboard satellite systems. Traditional smoothers require future observations to refine past state estimates, making them inherently suited for offline processing. However, real-time smoothing could be achieved through predictive algorithms that estimate future states based on current and past data. These algorithms use patterns from past data to make smart guesses about what the future data will look like. With these predictions, the algorithm can simulate

having future information and use it to make smoother, more accurate adjustments to the satellite's position in real time.

Bibliography

[1] W. Fehse. Autonomous Rendezvous and Docking of Spacecraft. First. Cambridge University Press, 2003 (cit. on pp. 1, 2).

[2] Crew Exploration Vehicle Docking With Iss 1. URL: https://sciencephotogallery.com/featured/1-crew-exploration-vehicle-docking-with-iss-nasawalter-myersscience-photo-library.html (cit. on p. 2).

[3] Steven J. Miller. The Method of Least Squares. Tech. rep. Williamstown, MA 01267: Williams College (cit. on p. 3).

[4] M. Meriem et al. C. Bouchra. Study of State Estimation Using Weighted-Least Squares Method (WLS). Tech. rep. Casablanca, Morocco: Higher National School of Electrical and Mechanical Engineering (ENSEM), 2016 (cit. on p. 3).

[5] P. Stoica and K. C. Sharman. «Maximum Likelhood Methods for Direction of Arrival Estimation». In: IEEE transactions on acoustic, speech and signal processing 38 (July 1990), pp. 1132–1143 (cit. on p. 3).

[6] J. Zhengk F. Cui D. Gao. «Magnetometer-Based Orbit Determination via Fast Reconstruction of Three-Dimensional Decoupled Geomagnetic Field Model». In: Journal of Spacecraft and Rockets 5 (2021), pp. 1374–1386 (cit. on pp. 5, 14).

[7] K.J.C. den Hollander. «Modeling and Characterizing a Sun Sensor for Nano Satellites». Elect. Eng. thesis. Enschede, The Netherlands: University of Twente, Feb. 2017 (cit. on p. 5).

[8] F. L. Markley and J. L. Crassidis. Fundamentals of Spacecraft Attitude Determination and Control. Springer and Microcosm Press, 2014 (cit. on pp. 6, 13, 32–34, 36).

[9] MAG-3. URL: https://www.satcatalog.com/component/mag-3/ (cit. on p. 6).

[10] Explorer GPS Receiver. URL: `https://gdmissionsystems.com/products/communications/spaceborne-communications/spaceborne-gps-receivers/explorer-gps-receiver` (cit. on p. 6).

[11] AQUILA-DO2 sun sensor. URL: `https://satsearch.co/products/newspace-systems-nfss-411-fine-sun-sensor` (cit. on p. 6).

[12] J. Steinbring and U. D. Hanebeck. S2̂KF: The Smart Sampling Kalman Filter. Tech. rep. Germany: Karlsruhe Institute of Technology (KIT) (cit. on p. 7).

[13] K. Fuji. Extended Kalman Filter. Tech. rep. The ACFA-Sim-J Group (cit. on p. 7).

[14] D. Jian-min L. Dan et al. «An Improved FastSLAM Algorithm for Autonomous Vehicle Based on the Strong Tracking Square Root Central Difference Kalman Filter». In: IEEE International Conference on Intelligent Transportation Systems. Beijing, China, 2015, pp. 693–698 (cit. on p. 7).

[15] A. Dey S. Chatterjee A. Chowdhury. DDF for Power Systems Dynamic State Estimation. Tech. rep. Durgapur, India: Department of Electrical Engineering NIT Durgapur (cit. on p. 7).

[16] A. Paul and E. A. Wan. «RSSI-Based Indoor Localization and Tracking Using Sigma-Point Kalman Smoothers». In: IEEE journal of selected topics in signal processing 3 (Oct. 2009), pp. 860–873 (cit. on pp. 7, 17).

[17] F. Gustaffson. «Particle Filter Theory and Practice with Positioning Applications». In: IEEE AE Systems Magazine 25 (July 2010), pp. 53–81 (cit. on p. 8).

[18] Crew Exploration Vehicle Docking With Iss 1. URL: `https://sciencephotogallery.com/featured/1-crew-exploration-vehicle-docking-with-iss-nasawalter-myersscience-photo-library.html` (cit. on p. 9).

[19] Fuse Sensor Data to Improve Localization. URL: `https://www.theconstruct.ai/robotigniteacademy_learnros/ros-courses-library/ros-robot-localization-package/` (cit. on p. 9).

[20] L. Cheng S. Xiao et al. «An adaptive identifcation method for outliers in dam deformation monitoring data based on Bayesian model selection and least trimmed squares estimation». In:
Journal of Civil Structural Health Monitoring 14 (Dec. 2023), pp. 763–779 (cit. on pp. 9, 22).

[21] C. D. Karlgaard. «Nonlinear Regression Huber–Kalman Filtering and Fixed-Interval Smoothing». In: Journal of guidance, control, and dynamics 38 (Feb. 2015), pp. 322–330 (cit. on pp. 9, 21, 25).

[22] PRISMA. URL: `https://www.eoportal.org/satellite-missions/prisma-prototype#spacecraft` (cit. on p. 10).

[23] $B20_thruster$. URL:
`https://www.satcatalog.com/component/b20-thruster/` (cit. on p. 11).

[24] What is an Electric Propulsion System for Satellites? URL:
`https://www.satnow.com/community/what-is-an-electric-propulsion-system-for-satellites` (cit. on p. 11).

[25] J. Stark P. Fortescue G. Swinerd. Spacecraft Systems Engineering. Fourth. Wiley, 2003 (cit. on p. 11).

[26] W. J. Larson J.R. Wertz. Space Mission Analysis and Design. Third. Springer Dordrecht, 1999 (cit. on p. 12).

[27] C. Fraser and S. Ulrich. «An adaptive Kalman Filter for Spacecraft Formation Navigation using Maximum Likelihood Estimation with Intrinsic Smoothing». In: Annual American Control Conference. Wisconsin Center, Milwaukee, USA, June 2018, pp. 5843–5848 (cit. on p. 15).

[28] S. Särkkä. «Unscented Rauch–Tung–Striebel Smoother». In:
IEEE transactions on automatic control 53 (Apr. 2008), pp. 845–849 (cit. on pp. 16, 19).

[29] Unscented Kalman Filter and Smoother. URL:
`https://it.mathworks.com/help/fusion/ug/introduction-to-estimation-filters.html` (cit. on p. 16).

[30] Cross-covariance Matrix. URL:
`https://www.statlect.com/glossary/cross-covariance-matrix` (cit. on pp. 18, 20).

[31] C. Karlgaard. «Robust Adaptive Estimation for Autonomous Rendezvous in Elliptical Orbit». PhD thesis. Virginia Polytechnic Institute and State University, June 2010 (cit. on pp. 21, 25).

[32] L. Chang B. Zhu et al. «Huber-Based Adaptive Unscented Kalman Filter with Non-Gaussian Measurement Noise». In:
Circuits systems and signal processing (Sept. 2018) (cit. on pp. 23, 46).

[33] J. E. White R. R. Bate D. D. Mueller. Fundamentals of Astrodynamics. Dover Publications, 1971 (cit. on pp. 29, 30, 32).

[34] Orbital Mechanics and Astrodynamics. URL: `https://orbital-mechanics.space/classical-orbital-elements/perifocal-frame.html` (cit. on p. 31).

[35] Norbert Zehentner. «Kinematic orbit positioning applying the raw observation approach to observe time variable gravity». PhD thesis. Graz University of Technology, Nov. 2016 (cit. on p. 33).

[36] Victor Jungnell. «Guidance Methods for Earth Observation Satellites». PhD thesis. Royal Institute of Technology School of Engineering Sciences, Sept. 2012 (cit. on p. 36).

[37] M. J. Sidi. Spacecraft Dynamics and Control. Cambridge University Press, 1997 (cit. on p. 38).