UNIVERSITÀ DEGLI STUDI DI PADOVA

UNIVERSITA' DEGLI STUDI DI PADOVA

Dipartimento di Ingegneria Industriale

Corso di Laurea Magistrale in Ingegneria Aerospaziale

Unscented Kalman Filters for Vision-Based Relative Dynamics Estimation between Cooperating Spacecraft

Relatore: Prof. Enrico Lorenzini

Correlatore: Dott. Andrea Valmorbida

Laureando: Giovanni Rizzolo

Matricola: 1156711

Anno Accademico 2018/2019

“To infinity and beyond”

Buzz Lightyear, Toy Story

Abstract

Over the past few decades, autonomous spacecraft proximity operations have become a key technology for space missions. The growing autonomy level of spacecraft leads to design space missions characterized by new capabilities, more flexibility and more robustness. Especially the capability of autonomously estimating the relative dynamics between a Chaser satellite and a Target spacecraft flying in proximity is crucial in the fields of Formation Flight, Satellite Rendezvous and Docking, On-Orbit Servicing or Repairing and Active Removal of defunct satellite.

At the moment there are several operational satellites in orbit associated with various types of missions from the scientific studies to commercial communications. Each satellite requires a dedicated operational team that, from a ground station, monitors the spacecraft, controls it, faces and solves potential problems. Most of the control operations are repetitive and well understood tasks and therefore can clearly be made autonomous within the spacecraft.

In addition, when considering spacecraft flying in proximity, especially the Formation Flight, the number of satellites that need to be controlled can be great. This adds other complexity for the operational teams, in particular if the satellites need information regarding the relative dynamics with respect to the other spacecraft.

In this thesis, autonomous estimation of the relative dynamics between spacecraft flying in proximity is discussed and tested, focusing on the performances of the estimation algorithms, based on Kalman filters. The aim is to propose alternatives for improving these performances, taking advantages from the use of Unscented Kalman filters that are a more accurate version of the standard Kalman filters.

Kalman filters are a well-established solution for the definition of navigation algorithms. The two most widely used are the classical Kalman Filter, designed to address the estimation problem for linear systems, and the Extended Kalman Filter, designed for the non-linear estimation problems.

Employing the Unscented formulation of the Kalman filters, it is possible to increase the accuracy of the relative dynamics estimation, especially considering the relative rotational motion. Along with this benefit, however, there is an increase in the computational time of the navigation algorithm due to the operational complexity introduced with the new filters.

Three alternative formulations to the standard Kalman filters are presented in this thesis: the Unscented Kalman Filter (UKF) and two other variants of this, i.e. the Single Propagation Unscented Kalman Filter (SPUKF) and the Extrapolated Single Propagation Unscented Kalman Filter (ESPUKF).

Following tests and comparisons, the new proposed solutions appear as valuable alternatives to the standard solutions offering good compromises between estimation accuracy and computational time.

Moreover, these filters appear robust with respect to realistic conditions, hence they maintain good estimation accuracy. The evaluated realistic conditions concern the measurements coming from an actual sensor, a time delay on the measurements input to the filters and uncertainties or noises on the other data needed by the filters.

These considerations are the outcome of numerical simulations and sensitivity analyses carried out in MATLAB using experimental measurements collected during specific tests conducted in the SPARTANS facility of the University of Padova.

Sommario

Negli ultimi decenni, le operazioni autonome di prossimità tra i satelliti sono diventate una tecnologia chiave per le missioni spaziali. Il costante aumento dei livelli di autonomia dei satelliti porta alla possibilità di progettare missioni caratterizzate da nuove capacità, più flessibilità e maggiore robustezza. In special modo risulta cruciale la capacità di stimare autonomamente la dinamica relativa tra un satellite Inseguitore e uno Bersaglio quando si considerano i campi del volo in formazione, dei Rendez-vous e agganci tra satelliti, del loro rifornimento o riparazione in orbita e della rimozione attiva di satelliti defunti.

Il numero dei satelliti operativi in orbita, associati a vari tipi di missioni, dagli studi scientifici alle comunicazioni commerciali, è in costante crescita. Ogni satellite necessita di un team operativo dedicato che, da una stazione a terra, monitora il satellite, lo controlla, affronta e risolve potenziali problemi. La maggior parte delle operazioni di controllo sono attività ripetitive e ben note che perciò possono chiaramente essere rese autonome all'interno del satellite stesso.

Considerando i satelliti che volano in prossimità tra loro, specialmente per il volo in formazione, il numero di satelliti che si devono controllare può diventare elevato. Ciò comporta ulteriore complessità per i team operativi, in particolare se i satelliti necessitano anche di informazioni riguardanti la dinamica relativa tra loro.

In questa tesi si discuterà della stima autonoma della dinamica relativa tra satelliti in volo in prossimità, concentrandosi sulle prestazioni degli algoritmi di stima basati sui filtri di Kalman. Lo scopo è quello di proporre nuove alternative per migliorare queste prestazioni sfruttando i filtri di Kalman Unscented che sono versioni più evolute e precise dei filtri di Kalman standard.

I filtri di Kalman sono una soluzione ben stabilità per la definizione degli algoritmi di navigazione. I due più diffusi sono il filtro di Kalman classico e il filtro di Kalman Esteso, formulati rispettivamente per la stima dei moti relativi traslazionale e rotazionale.

Sfruttando la formulazione Unscented dei filtri di Kalman è possibile incrementare la precisione della stima della dinamica relativa, specialmente considerando il moto d'assetto relativo. Assieme a questo beneficio, comunque, ci sarà anche un aumento del tempo computazionale per l'algoritmo di navigazione, dato che questi nuovi filtri introducono una maggiore complessità operativa.

Le alternative presentate in questa tesi sono tre: il filtro di Kalman Unscented (UKF) e due sue varianti, cioè il filtro Unscented a singola propagazione (SPUKF) e quello a singola propagazione estrapolata (ESPUKF).

Le nuove soluzioni proposte risultano essere delle valide alternative rispetto alle soluzioni standard, infatti offrono dei buoni compromessi tra precisione della stima e tempo di calcolo.
Note le prestazioni dei filtri in quanto ad accuratezza di stima e tempo di calcolo, è possibile determinare la soluzione migliore per l'algoritmo di navigazione in base ai requisiti di missione.

Inoltre, questi filtri risultano robusti quando si considerano condizioni realistiche, infatti mantengono una buona accuratezza di stima. Le condizioni realistiche considerate interessano tutte le informazioni fornite ai filtri che sono raccolte da un sensore reale e associate ad un ritardo temporale oppure sono associate a delle incertezze o a dei rumori.

Queste considerazioni sono il risultato di simulazioni numeriche e di analisi di sensitività svolte in MATLAB sfruttando misure sperimentali raccolte tramite test specifici condotti nel laboratorio SPARTANS dell'Università degli Studi di Padova.

Contents

Chapter 1

Introduction

1.1 Problem Definition

Nowadays, Distributed Space Systems, such as Spacecraft Formation, are becoming more and more interesting and adopted for future space missions. A Distributed Space System is a cooperative infrastructure formed by multiple satellites, where each of them can accomplish independently different operations. This mission design allows several benefits, with respect to missions consisting of a single large spacecraft, such as lower costs, better performances and more robustness to failures, since a single failed satellite can be easily replaced by another one or its operations can be distributed among the remaining satellites.

The Formation Flight, however, requires new strategies and technologies. The most demanding issue concerns the Global Navigation and Control system that becomes highly complex. Controlling from the ground all the satellites flying in a formation is hardly possible, so it is required that the satellites manage autonomously their own navigation. Providing a high level of autonomy to the spacecraft in a formation is very challenging, especially for the estimation of the relative dynamics between each satellite.

Satellites belonging to a formation can be small and in these cases their resources are limited in terms of mass, volume, power consumption and computational capacity. For these reasons, monocular or stereo cameras are the most widely used sensors to provide measurements of the relative pose between the satellites flying in proximity, thanks to their low complexity and power consumption.

Indeed, a reliable strategy for filtering the measurements collected by these instruments is required in order to obtain an accurate estimation of the relative dynamics between the spacecraft.
For space missions, Kalman filters are already a well-established solution for the relative dynamics estimation problem.

There are several current projects dealing with relative dynamics estimation and they are all based on the standard Kalman filters focusing on new instruments to measure the relative pose [9] or on new strategy for facing the estimation problem with non-cooperative spacecraft [8] [10].

In this thesis we focused on the filters of the navigation algorithm developing and comparing different type of Kalman filters in order to evaluate their performances in terms of accuracy of the relative dynamics estimation, the required computational time and their complexity.

1.2 Kalman Filters

The Kalman filter, first invented in the 1960 by Rudolph E. Kalman, has been widely used as a state estimation technique in numerous studies and applications in different fields, from economy to engineering.

The base Kalman Filter (KF) is dedicated to performing the state estimation for only linear systems. In order to study also highly non-linear systems, many variations of these filters have been developed. The most widely used for non-linear systems is the Extended Kalman Filter (EKF), which is employed also by the NASA

Ames Research Centre for estimating the dynamics of a space vehicle that is non-linear. This formulation of the Kalman filter takes advantage of a linearization of the system using a first-order Taylor series approximation that leads to a suboptimal solution of the non-linear estimation problem. The more non-linear is the studied system the more the solution provided by an EKF is far from the optimal solution.

To overcome this issue, for highly non-linear systems, other types of Kalman filter have been developed. Most of these are computationally complex and demanding, but one leads to very good results without excessive complexity: the Unscented Kalman Filter (UKF). This one is based on the use of the Unscented Transformation (UT) before the prediction step. The UT is a different approach that uses a deterministic sampling to predict the state vector and the error covariance. Thanks to this new approach the UKF is clearly more computationally demanding and time consuming than the KF and the EKF, but ensures an accuracy of at least the second-order Taylor series approximation even for highly non-linear systems.
At a later time, in order to reduce the computational time and upgrade the precision of the UKF, other approaches were developed. Only few of these succeed at speeding up the filter without losing all its accuracy or at upgrading its accuracy without highly increasing the computational time. The two most known and used approaches are the ones that leads to the Single Propagation Unscented Kalman Filter (SPUKF) and the Extrapolated Single Propagation Kalman Filter (ESPUKF).

The Unscented Kalman filters are used and analyzed in several current projects to evaluate their performances for the position and attitude tracking of a re-entry vehicle [12] or for a launch vehicle [13].

In this study these filters are used for the estimation of the relative pose and dynamics between the SPARTANS (see later) spacecraft unit and the chaser unit represented by a monocular camera placed in a fixed frame. The attitude and the translational motion are independent from each other and so they are analyzed separately:

- For the translational motion the filters used are: the KF, the UKF, the SPUKF and the ESPUKF;
- For the attitude motion, which is non-linear, the filters used are: the EKF, the UKF, the SPUKF and the ESPUKF.

In a real-time application the two motions are analyzed simultaneously, and the global navigation estimator process consists of two filters running in parallel, as shown in Fig. 1.1.

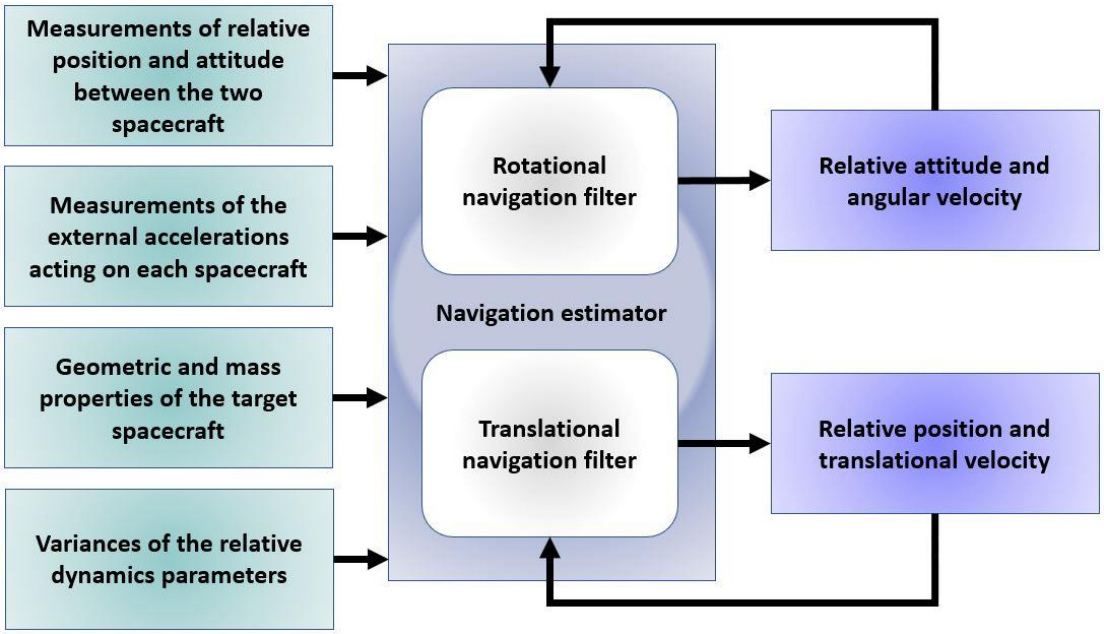

Figure 1.1: Conceptual outline of the navigation estimator based on parallel filters.

In order to fulfil the estimation of the relative dynamics, the filters need various information:

- The measurements of the relative pose between the spacecraft are needed for update and correct the propagation of the filters.
- The external accelerations acting on the spacecraft are necessary for the right evaluation of its motion.
- The geometric and mass properties of the target are needed for evaluating correctly the process model inside the filters.
- The variances of the relative dynamics parameters are necessary for the right setting of the filters.

The results obtained at every time instant are used by the filters to estimate the results of the next time instant obtaining a continuous evaluation of the relative dynamics.

The measurements of the relative pose were collected by a monocular vision camera to simulate the instruments mounted on actual spacecraft.

In addition, the spacecraft are considered cooperating, so they provide each other the measurements of the external accelerations acting on them and the values of their moments of inertia.

1.2 Thesis Outline

Before analyzing the performances of each filter formulation, all the information needed by them has to be defined, specifically:

- The measurements of the relative pose between the spacecraft were provided carrying out some tests with the SPARTANS facility of the University of Padova. In these tests two type of measuring systems are involved: a Motion Capture system, to define a fiducial reference, and a monocular vision camera, to simulate a possible instrument on board the satellite.
- From these measurements it was possible also to calculate the external accelerations acting on the spacecraft simulator employing some mathematical operations.
- The geometric and mass properties of the target satellite were analyzed developing a 3D model in SOLIDWORKS.
- The variances of the measurements provided to the filters were calculated through Montecarlo analyses.

Starting from a brief description of the SPARTANS facility and of the main instruments employed, these preliminary operations will be illustrated.
After this, the filters models will be discussed focusing on the differences between them. Then the process equations, used to analyze the motions of the spacecraft, will be defined and, basing on these, the filters will be set up for the specific cases of the relative dynamics estimation.

To conclude, the developed filters will be employed to estimate the relative dynamics between the spacecraft simulators of the SPARTANS facility and their performances, in terms of accuracy and computational time, will be compared. In addition, to test the behavior of these filters for more realistic conditions, some sensitivity analyses will be carried out.

Chapter 2

The SPARTANS Facility

The facility SPARTANS (Fig. 2.1), developed at the Center of Studies and Activities for Space (CISAS) "G. Colombo" and the Department of Industrial Engineering (DII) of the University of Padova, is a ground-based spacecraft proximity operation simulator. This facility has been designed as a testbed for various researches in the fields of Spacecraft Formation Flight (SFF), Rendez-Vous and Docking (RVD) and any other on-orbit proximity operations.

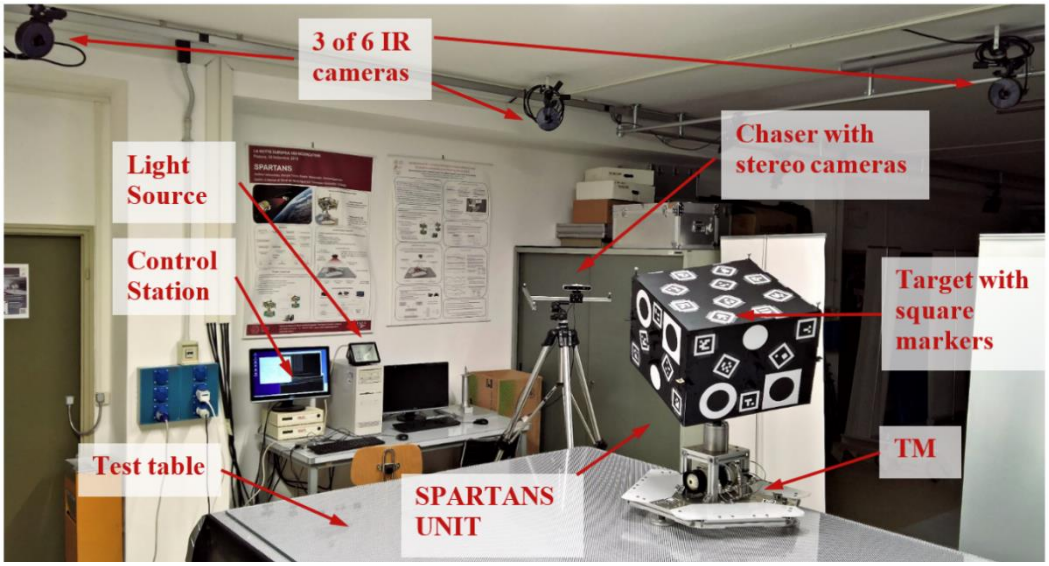

Figure 2.1: Set-up of the SPARTANS facility [7].

The SPARTANS facility allows to reproduce in laboratory the dynamics of two or more satellites flying on-orbit in close proximity, and it can be employed for the development and the verification on the ground of algorithms for relative attitude and position Guidance, Navigation and Control (GN&C) (Fig. 2.2).

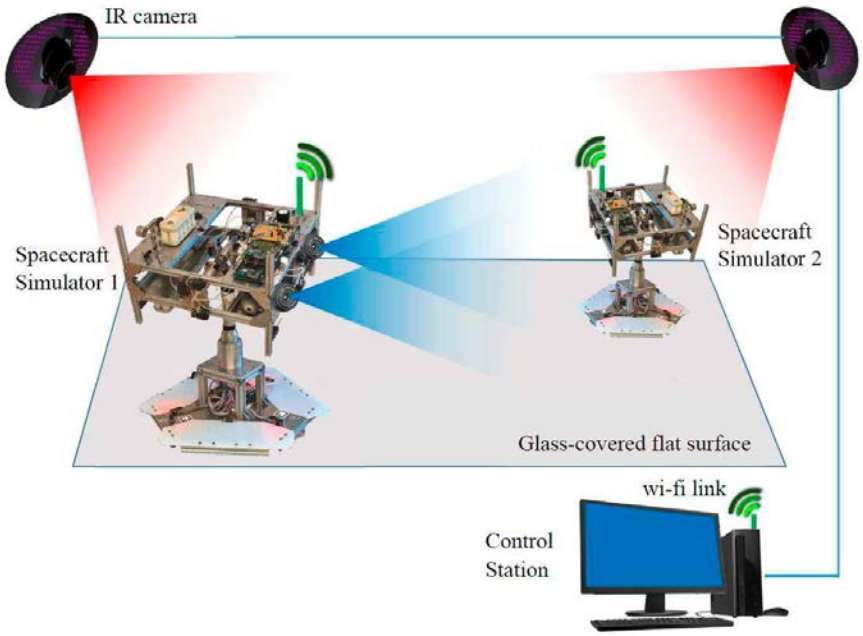

Figure 2.2: Reproduction of a proximity motion simulation [5].

The core of this facility are the spacecraft simulators, which represent the units of an on-orbit formation. Each simulator is composed of two main parts, an upper one called the Attitude Module (AM), characterized by three rotational degrees of freedom, and a lower one called the Translational Module (TM), characterized by two planar translational degrees of freedom. The three rotational degrees of freedom of the AM are enabled by a three-joint mechanical system that provides triaxial low-friction rotation, taking advantage of low-friction spherical bearings. The rotation around the yaw axis, the one perpendicular to the test-table, is completely free, while the rotations around the other two axes are limited between ± 40°.
To provide the two planar degrees of freedom to the TM, the simulators are placed on a glass-covered testing table and compressed air is blown from the base of the TM towards the table forming an air cushion that prevents the contact between the two bodies. In this way the spacecraft can move with very low friction on the table gaining the two planar degrees of freedom.
Thanks to these five degrees of freedom, it is possible to simulate several relative motions between spacecraft. This allows to collect measurements of different maneuvers for a wide spectrum of cases.

At the time of the writing this thesis, one spacecraft simulator is fully operative, while a second one is under development and cannot operate on its own, so it is used to simulate a casual motion applying maneuvers by hand. An external fixed frame hosting two cameras (Fig. 2.3), a ZED camera produced by STEREOLABS and a DUO camera produced by Code Laboratories, is employed to simulate a chaser spacecraft that observes the simulators and can be used to test algorithms and systems for Vision-Based Relative Navigation. Between the two available cameras, only the ZED camera was used to collect measurements, since the focus of the thesis is the performances' analysis of the navigation filters rather than the analysis of the measuring instruments.

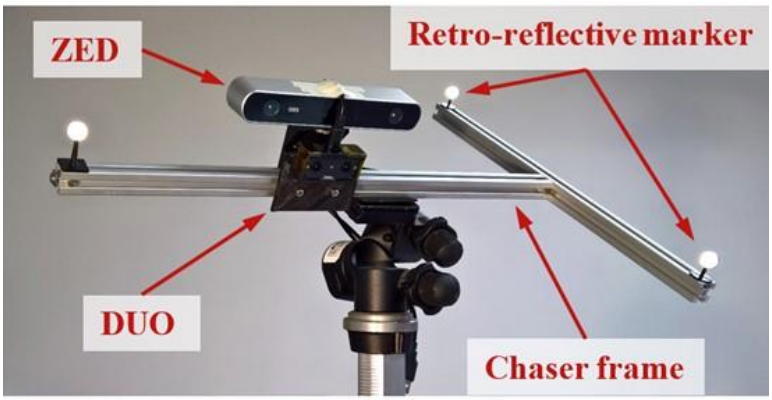

Figure 2.3: The Chaser frame and the two hosted vision cameras [7].

A global navigation system is implemented by a Motion Capture system to provide precise measurement of position and attitude of the spacecraft simulators. This system is composed of a set of six infrared digital cameras, some retro-reflective markers and a computer that collects and processes the measurements taken by the infrared cameras.

The SPARTANS facility is completed by a remote-control station that controls the simulators and the vision cameras and receives from them the information of telemetry and housekeeping data as well as the collected images.

This facility has different advantages: it allows to simulate five degrees of freedom motion, the costs of every test is contained, the duration of a test is limited only by the amount of propellant, i.e. compressed air, that can be stored on-board the modules, the interface of the simulator with the operator is user friendly.

2.1 Motion Capture System

The Motion Capture system, developed by BTS Engineering, is composed of six Sony XC-75 infrared cameras (Fig. 2.4), with a resolution of 768(H)x494(V) pixel, placed around the testbed where the satellite simulators moves, and of a software workstation, for launch and control the acquisition and analyze in post-processing the collected data.

Figure 2.4: One of the infrared cameras of the Motion Capture system.

The six infrared cameras have an acquisition frequency of 50 Hz and can detect the position of an infrared spherical marker with an uncertainty of half a millimeter [4]. To determine the 3D position in space of a marker it has to be in the field of view of at least two cameras. The fields of view of the six cameras overlap the operative volume above the testbed; in this way there are redundant measurements for every marker, leading to a gain in accuracy. Thanks to this accurate system, the measurements taken by these cameras are used as a fiduciary reference to evaluate the other measurements taken with the monocular vision camera and to evaluate the performances of the developed filters in estimating the relative motion between two spacecraft.

The software workstation, composed by a computer and two frame grabbers connected to the cameras, allows to collect and elaborate the images taken from the cameras in order to extract the measurements needed to determine the position and the orientation of the spacecraft simulator.
Some components of this system can be seen in Fig. 2.1.

This system detects and tracks, during the post-processing, the position of all the infrared retro-reflective markers. Three of these markers are on the corner of the top surface of the attitude module, the target spacecraft, and allow to follow its movements; while other four retro-reflective markers are placed in the structure that support the vision cameras, which represent the inspector satellite, in order to evaluate the position and the orientation of the inspector's reference frame. Before starting an acquisition of the motion of the spacecraft with these cameras, a global reference frame for the Motion Capture system must be defined in order to express the measurements with respect to a fixed coordinate system. Then, all the infrared cameras have to be calibrated for the space where the module will be moving later. For this purpose, a set of three orthogonal sticks with different infrared markers has been made up in order to form the three axes of the global reference frame; then using one of these sticks and moving it in the planned moving space of the module, it is possible to calibrate the Motion Capture system.

Once an acquisition of the motion of the satellite is over, all these markers are manually numbered and connected between them during the post-processing phase. At the end of all this, the result of the acquisition is the position of all the marker at every time instant referred to a global reference frame, defined before the acquisition.

The results are written into a text file that can be read by the software MATLAB to develop and compare the filters performances for the navigation algorithm.

Using this data, it is possible to determine the position and the orientation of the reference frame of the fixed structure, that represent the reference frame of the chaser satellite, and the reference frame of the target attached to the attitude module, both referred to the global reference frame.

2.2 Monocular vision camera

The monocular vision camera, used for the acquisition of the relative motion between the target and the chaser spacecraft, is the left camera of a stereo rig, the ZED camera (Fig. 2.3), not employed in its stereo form in this thesis. This is a monocular camera that collect color images with a resolution of 4 Megapixel. It has an electronic synchronized rolling shutter. Its focal length is 2.8 mm and its field of view is 90 deg (H) x 60 deg (V). During the acquisitions this camera is set to collect images with a frequency of 10 Hz.

All the settings and the controls are sent to this instrument using another computer workstation that host a LINUX operative system and that is connected to the camera via USB. In this way it is possible to set the camera acquisition frequency and even the resolution of the acquired images.

This monocular camera has been calibrated acquiring several images of a calibration chessboard and analyzing them employing the camera calibration toolbox in MATLAB. In this way it was possible to estimate the intrinsic parameters of the camera.
Knowing these parameters, the images collected from the camera during a relative motion simulation are analyzed in order to estimate position and attitude of the target module with respect to the camera.

A set of 30 square markers are placed on the external black shell of the attitude module (Fig. 2.5). In order to determine the position and the orientation of the spacecraft these square markers have to be detected during the analysis of the images. Then, the detected markers have to be identified. To do so, the used markers belong to an ARUCO codification family, therefore, each one has its own defined features. After the identification of the markers on one image, it is possible to reconstruct the complete map of the markers from which the relative pose between the target module and the camera can be estimated.

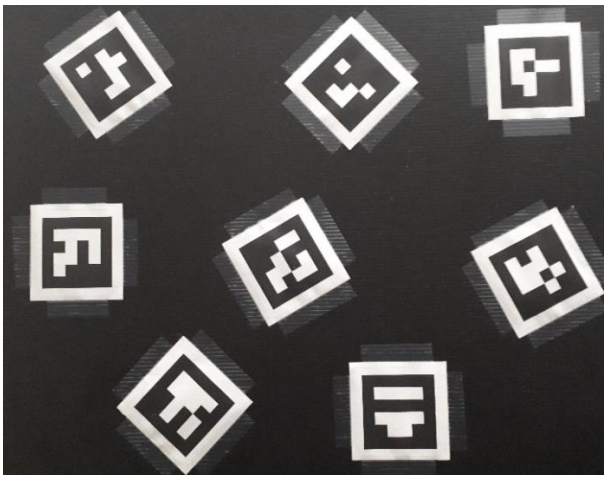

Figure 2.5: Some of the ARUCO markers placed on the external surface of the target spacecraft.

The images analysis is developed using a specific program created by the ARUCO Company, which detects, identifies and extracts the edges of every square marker in the image and, consequently, creates a text file where the coordinates of the edges of all the detected marker for every image are written.
Later in MATLAB, it is possible to use the information from this file to reconstruct the position and the orientation of the ARUCO markers for every image.

Chapter 3

Filters' input definition

The Kalman filters need various input data in order to work properly. For the relative dynamics estimation, the information needed concern the relative pose between the two satellites, the external acceleration acting on the target spacecraft, its moments of inertia and, lastly, the variances of all the parameters that define the relative dynamics. To determine this information, we have carried out the following operations and analyses.

3.1 Relative pose measurements

During the simulations, the presented instruments, the MC system and the monocular camera, collects measurements of the relative position and attitude of the target spacecraft with respect to the chaser fixed frame. These measurements are not directly comparable, since they refer to diverse reference frames. Therefore, they have to be made coherent using proper reference frame transformations determined during a dedicated calibration procedure.

All the reference frames and the roto-translational matrices, needed for the reference frame transformations, are shown in Fig. 3.1.

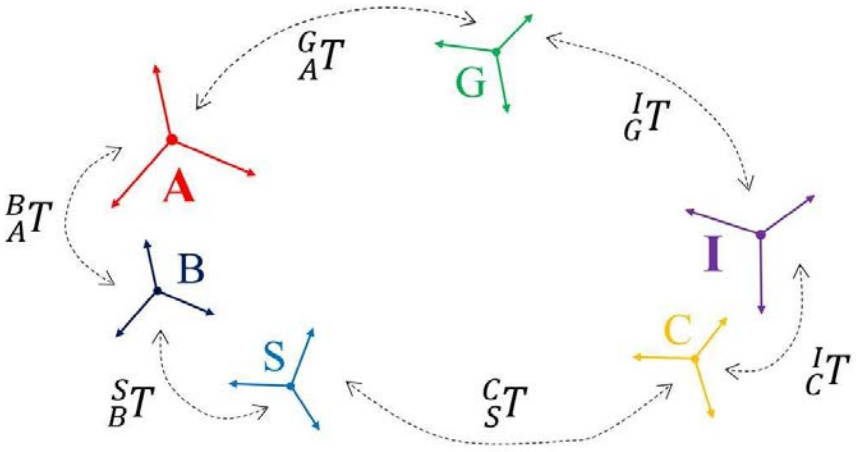

Figure 3.1: Reference frames and their relative transformation matrices [4].

The reference frames considered for the measurements correction are the following:

- **G**: It is the Global reference frame (Fig. 3.2). This is a Local Vertical – Local Horizontal (LVLH) frame of reference defined as explained before using a set of three perpendicular sticks with retro-reflecting markers. All measurements collected by the Motion Capture are referred to this reference frame.

- **A**: It is the Attitude module reference frame (Fig. 3.2). This is defined starting from the position of the three retro-reflective markers on board of the module. The origin $\mathbf{O_a}$ of this frame is coincident with the central marker, $\mathbf{M_0}$. The three axes are defined using the position of the three markers at every instant. First the unit vectors are determined from these equations:

$$\mathbf{p} = \frac{\mathbf{M_1} - \mathbf{M_0}}{\|\mathbf{M_1} - \mathbf{M_0}\|_2} \qquad q = \frac{M_2 - M_0}{\|M_2 - M_0\|_2}$$

  Then it is possible to calculate the axes of this reference frame as follow:

$$\boldsymbol{x}_a = \mathbf{p} \quad ; \quad \mathbf{z_a} = \frac{\mathbf{p} \times \mathbf{q}}{\|\mathbf{p} \times \mathbf{q}\|_2} \quad ; \quad \boldsymbol{y}_a = \boldsymbol{z}_a \times \boldsymbol{x}_a$$

- **B**: It is the spacecraft Body reference frame (Fig. 3.2). Its origin $\mathbf{O_b}$ is coincident with the center of mass of the attitude module, and center of rotation of the mechanical joint, while its axes are parallel to those of the **A** reference frame.

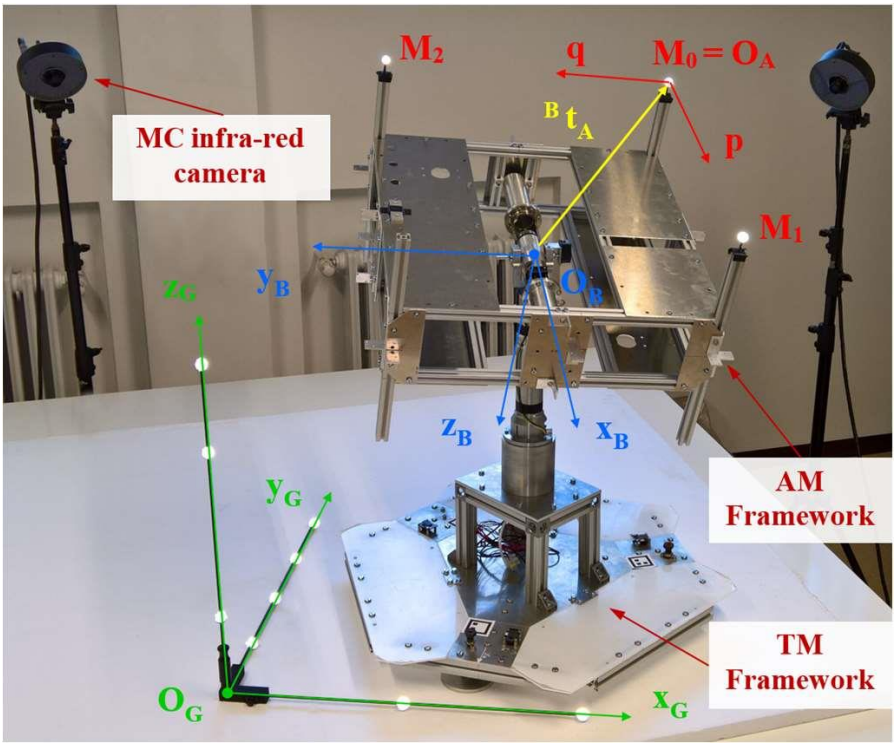

Figure 3.2: Global, Attitude module and Body reference frames [4].

- **S**: It is the Square markers reference frame (Fig. 3.3). This is defined by the map of the square markers attached to the external black shell of the attitude module. Its origin $\mathbf{O_s}$ is placed at the geometrical center of the marker with ID 0. The axis $\mathbf{z_s}$ is perpendicular to the marker plane while the axes $\mathbf{x_s}$ and $\mathbf{y_s}$ are parallel to the marker edges.

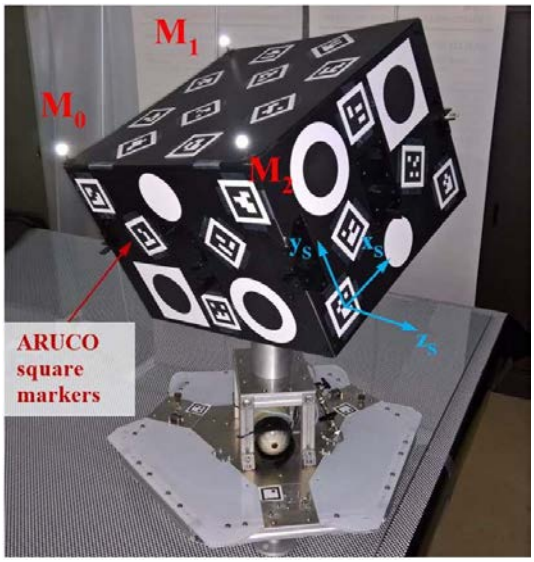

Figure 3.3: Square marker reference frame [4].

- **C**: It is the Camera reference frame (Fig. 3.4). This one has its axis $\mathbf{z_c}$ aligned with the optical axis of the monocular camera while the other two axes, $\mathbf{x_c}$ and $\mathbf{y_c}$, belong to the focal plane of the camera.

- **I**: It is the Inspector reference frame (Fig. 3.4). This is defined by the retro-reflective markers placed on the fixed frame that support the camera and that represent the Inspector in this case. The procedure for defining this reference frame is similar to the one followed to determine the Attitude module reference frame. Its origin $\mathbf{O_i}$ is coincident with the marker $\mathbf{M_4}$, then the unit vectors **p** and **q** are defined respectively by the couple of markers $\mathbf{M_3}$ **-** $\mathbf{M_4}$ and $\mathbf{M_5}$ **-** $\mathbf{M_4}$.

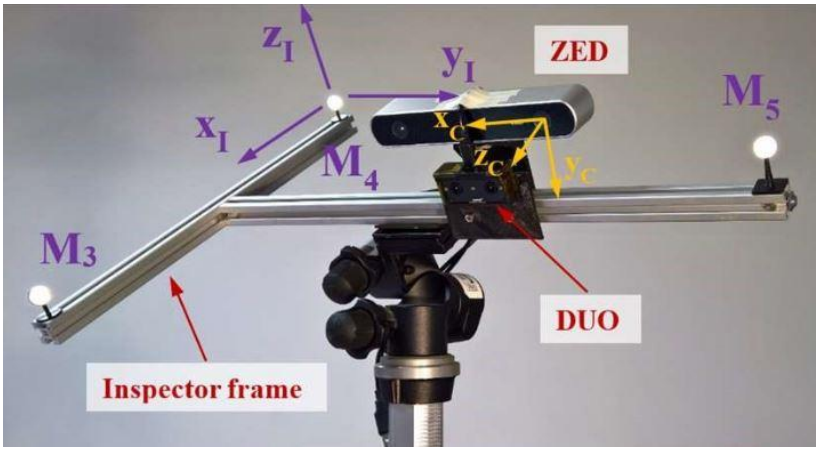

Figure 3.4: Camera and Inspector reference frames [4].

Employing the measurements collected, during an acquisition, it is possible to express three roto-translational matrices, the one between **A** and **G** reference frames and the one between **F** and **G** reference frames, thanks to the Motion Capture, and the one between **S** and **C** frames, thanks to the monocular camera.

The other three roto-translational matrices, $\boldsymbol{{}^{I}_{C}T}$, $\boldsymbol{{}^{S}_{B}T}$, $\boldsymbol{{}^{A}_{B}T}$, are constant and were estimated through a set of specific calibration procedures. The estimation of the roto-translational matrix between **I** and **C** reference frames started taking measurement, with both Motion Capture system and monocular camera, of various fiducial retro-reflective spherical markers placed on the test-table. The Motion Capture system measured the 3D positions of this markers and also the 3D positions of the three retro-reflective markers placed on the Inspector supporting structure. The camera acquired images of the fiducial markers that were later analyzed with a dedicated software developed in MATLAB in order to detect the markers and determine their 2D positions on the camera image plane. Then, a non-linear Least-Square Problem was solved in MATLAB using the Levenberg-Marquardt algorithm to find the roto-translational matrix $\boldsymbol{{}^{I}_{C}T}$ that minimizes the reprojection errors of the 3D positions of the fiducial markers on the image plane.

The estimation of the other two matrices required a different calibration procedure considering that, during any pure rotational motion of the Attitude module, the center of each retro-reflective spherical marker, on board of the module, and the center of the ARUCO marker with ID 0, that is coincident with the origin of the **S** reference frame, move on spheres whose center is coincident with the center of rotation of the mechanical gimbals. To start this calibration, the Attitude module was placed in various static positions and for every position the camera took images of the module, while the Motion Capture measured the 3D coordinates of the three spherical markers on board the module. These collected data were later used in a non-linear Least-Square Problem to find the radii of each sphere and the 3D coordinates of the origin $\mathbf{O_b}$ expressed in the two reference frames **A** and **S**. in this way it was possible to find the roto-translational matrices $\boldsymbol{{}^{S}_{B}T}$ and $\boldsymbol{{}^{A}_{B}T}$.

At this point, with the knowledge of these last three constant matrices, it is possible to reconstruct and to make coherent the relative motion between the two spacecraft, the Inspector and the SPARTANS module, starting both from the measurement taken with the monocular camera and from those taken with the Motion Capture system.

Finally, these two relative motions can be compared to verify their coherence. From this comparison it is most likely to find out that the two estimated motions are indeed coherent, but that they are shifted in time as shown in Fig. 3.5.

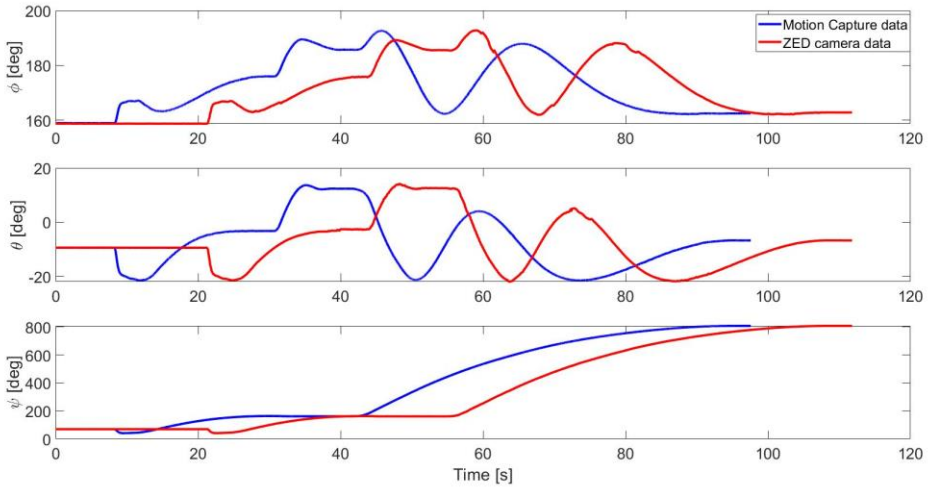

Figure 3.5: Time-shift between the measurements of the attitude of the module during a rotational motion.

This shift is due to the different instant in which the two kind of instruments start their acquisitions. This issue was overcome by solving a non-linear Least-Square Problem that gives as output the value of the time shift that minimizes the error between the curves of the estimated motions. Applying the found time shift to the time vector of one set of data it is possible to synchronize the results (Fig. 3.6).

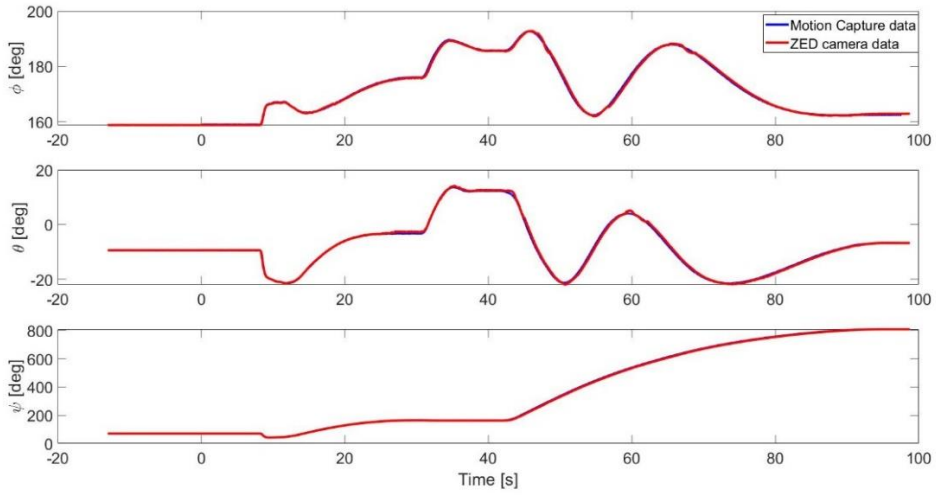

Figure 3.6: Measurements of the attitude of the module after the time-shift correction.

Another possible issue between the two set of results could have come up if the used instruments followed different clocks when collecting the images and writing down their catching-times. As clearly seen in Fig. 3.6, this issue appears not to happen because the clocks of the instruments are similar and because the acquisitions are not long enough for some mismatch to become relevant.

3.2 External accelerations

Starting from the collected measurements of the relative position and attitude, it is possible to extract the values of the external accelerations acting on the target spacecraft.

In order to reduce the errors and the noise of the measurements a suitable interpolation was made by using the "*smoothing spline*" feature in MATLAB. Without this smooth interpolation the velocity and mostly the calculated acceleration were highly affected by noise, due to the double derivative needed to obtain the acceleration from the measurement of the position.
To suppress the noise, it is enough to use a smoothing parameter of 0.99 (out of 1). The effect of this smooth interpolation can be clearly seen in Fig. 3.7, where attitude, angular velocity and angular acceleration with and without the smoothing are compared.

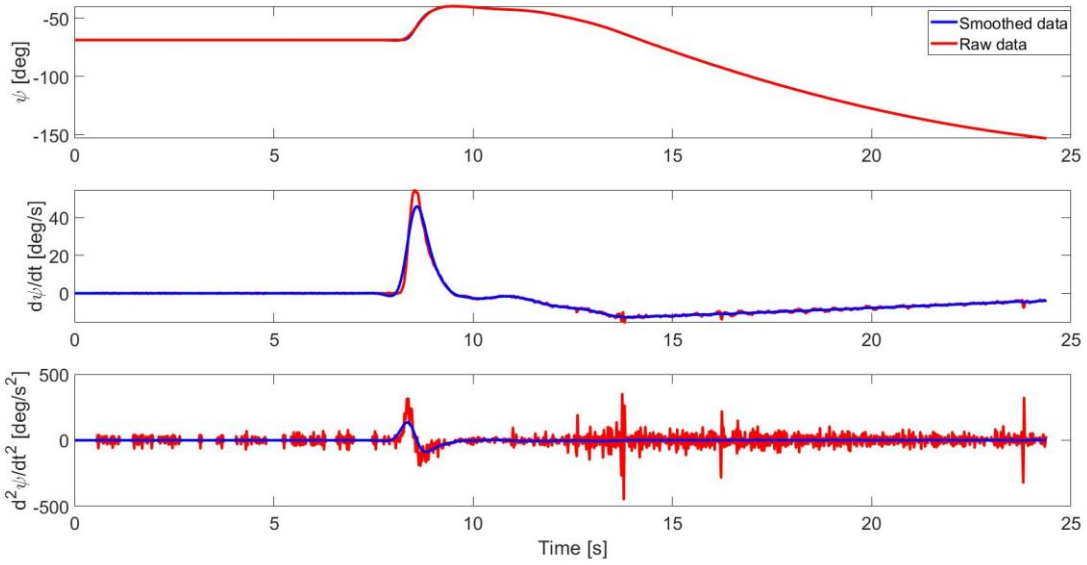

Figure 3.7: During this acquisition the module was moved by hand to simulate a random impulsive motion.

The translational velocity and acceleration are directly calculated with the derivation of the measurement of the relative position. On the other hand, the evaluation of the rotational velocity and acceleration needs more operations and is described as follows.

The angular velocity displayed in Fig 3.7 is obtained with the derivation of the measurement of the angular position of the Attitude module and so these are the velocity of roll, pitch and yaw, $\boldsymbol{\omega_\phi}$, $\boldsymbol{\omega_\theta}$, $\boldsymbol{\omega_\psi}$, which are referred to an inertial reference frame, and they are not coincident with the body angular velocity, $\boldsymbol{\omega_x}$, $\boldsymbol{\omega_y}$, $\boldsymbol{\omega_z}$, which are referred to the body reference frame. Therefore, another reference frame transformation is needed. To transform the inertial velocity into the body ones, three rotations has to be performed. The sequence of rotations choose is the '*123*', where the first rotation is applied around the x axes, that belongs to the inertial reference frame, the second rotation is around an intermediate y axes, and the last rotation is performed around the z axes belonging to the body reference frame.

Each rotation can be expressed as an orthogonal matrix function of the respective Euler's angle that quantifies the rotation. The used rotational matrices are:

$$\boldsymbol{R_1(\phi)} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{c(\phi)} & \boldsymbol{s(\phi)} \\ \mathbf{0} & \boldsymbol{-s(\phi)} & \boldsymbol{c(\phi)} \end{bmatrix}$$

$$\boldsymbol{R_2(\theta)} = \begin{bmatrix} \boldsymbol{c(\theta)} & \mathbf{0} & \boldsymbol{-s(\theta)} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \boldsymbol{s(\theta)} & \mathbf{0} & \boldsymbol{c(\theta)} \end{bmatrix}$$

$$\boldsymbol{R_3(\psi)} = \begin{bmatrix} \boldsymbol{c(\psi)} & \boldsymbol{s(\psi)} & \mathbf{0} \\ \boldsymbol{-s(\psi)} & \boldsymbol{c(\psi)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

In this notation, **s** and **c** represent respectively the sine and the cosine of the angles.

Multiplying in the correct sequence these matrices it is possible to obtain the rotation matrix that transforms the calculated angular velocity into the body ones:

$$\boldsymbol{R_{123}(\phi,\theta,\psi)} \;=\; \boldsymbol{R_3(\psi)\, R_2(\theta)\, R_1(\phi)}\,.$$

From this the result is:

$$\boldsymbol{R_{123}} = \begin{bmatrix} \boldsymbol{c(\psi)c(\theta)} & \boldsymbol{c(\psi)s(\phi)s(\theta) + s(\psi)c(\phi)} & \boldsymbol{c(\psi)c(\phi)c(\theta) + s(\psi)s(\phi)} \\ \boldsymbol{-s(\psi)c(\theta)} & \boldsymbol{-s(\psi)s(\phi)s(\theta) + c(\psi)c(\phi)} & \boldsymbol{-s(\psi)c(\phi)c(\theta) + c(\psi)s(\phi)} \\ \boldsymbol{s(\theta)} & \boldsymbol{-s(\phi)c(\theta)} & \boldsymbol{c(\theta)c(\phi)} \end{bmatrix}$$

This is the rotational matrix that transforms the inertial frame to the body frame.

Applying the rotational matrices to the inertial angular velocity it is possible to calculate the body angular velocity:

$$\boldsymbol{\vec{\omega}_{BI}} = \boldsymbol{R_3(\psi)}\begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\dot{\psi}} \end{Bmatrix} + \boldsymbol{R_3(\psi)R_2(\theta)}\begin{Bmatrix} \mathbf{0} \\ \boldsymbol{\dot{\theta}} \\ \mathbf{0} \end{Bmatrix} + \boldsymbol{R_3(\psi)R_2(\theta)R_1(\phi)}\begin{Bmatrix} \boldsymbol{\dot{\phi}} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix}$$

These operations can also be expressed in compact form as:

$$\vec{\omega}_{BI} = \begin{bmatrix} c(\psi)c(\theta) & s(\psi) & 0 \\ -s(\psi)c(\theta) & c(\psi) & 0 \\ s(\theta) & 0 & 1 \end{bmatrix} \begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix}$$

Performing this transformation, the inertial angular velocity is converted into the body angular velocity.

From the body angular velocity just calculated it is possible to obtain the total body angular accelerations that act on the attitude module $\boldsymbol{Acc_{Tot}}$. These accelerations consist of two components: the first one $\dot{\boldsymbol{\omega}}$ is due to the propagation of the angular velocity, defined by the Euler equation of dynamics for a free motion, and the second one $\boldsymbol{Acc_{Ext}}$ refers to the external accelerations, caused by the impulses, imposed manually, and by the friction, with the air and in the joint.

$$\boldsymbol{Acc_{Tot}} = \dot{\boldsymbol{\omega}} + \boldsymbol{Acc_{Ext}}$$

The Kalman filters have the capacity of estimating the first component of the total angular accelerations, $\dot{\boldsymbol{\omega}}$, therefore they are suitable for the estimation of the relative rotational dynamics between two spacecraft when their relative motion is unforced.
Considering a forced motion where there is the presence of perturbing angular accelerations, to pursue the estimation the filters need the input of these quantities since these are associated to external forces that cannot be predicted by the filters.

During the simulations carried out employing the SPARTANS facility, the target module was moved by hand and was affected by external disturbances caused by the friction with the air and into the mechanical joints; therefore, the simulated motions are clearly perturbed.

In order to evaluate even the angular velocity estimation performances of the filters, measurements of the external angular accelerations have to be provided, therefore the two components of the total angular accelerations must be separated.

To pursue this aim, it is possible to calculate the value of the $\dot{\boldsymbol{\omega}}$ at every time instant using the Euler's equation of dynamics without considering the external forces, employing the body rotational velocity and the calculated moments of inertia of the attitude module and then subtracting this vector from the global acceleration.

$$\begin{cases} I_x\dot{\omega}_x + (I_z - I_y)\omega_y\omega_z = \tau_x \\ I_y\dot{\omega}_y + (I_x - I_z)\omega_x\omega_z = \tau_y \\ I_z\dot{\omega}_z + (I_y - I_x)\omega_x\omega_y = \tau_z \end{cases}$$

Considering a free motion without the action of external forces, the angular accelerations due only to the propagation of the angular velocity can be calculated from the equations:

$$\begin{cases} \dot{\omega}_x = \dfrac{(I_y - I_z)\omega_y\omega_z}{I_x} \\ \dot{\omega}_y = \dfrac{(I_z - I_x)\omega_x\omega_z}{I_y} \\ \dot{\omega}_z = \dfrac{(I_x - I_y)\omega_x\omega_y}{I_z} \end{cases}$$

Once calculated, this angular acceleration vector has to be subtracted from the global accelerations:

$$Acc_{Ext} = Acc_{Tot} - \dot{\omega}$$

In this way it is possible to evaluate the external angular accelerations acting on the module, in order to give them as input to the filters.

In a real case it will be sufficient to have gyros on board of the spacecraft that measure the rotational rates which can be manipulated to evaluate the total angular accelerations. In this case the filter's estimation of the component $\dot{\omega}$ is not useful since the evaluation of the relative angular velocity would be based only on the upcoming measurements of the angular acceleration and velocity itself.

Further in the analysis, some noises were added to the external accelerations to make these values similar to measurements taken by actual sensors in order to verify the influence of instrument's noise on the performances of the navigation filters.

3.3 Moments of inertia

In order to estimate the moment of inertia of the target spacecraft, one SPARTANS module in this case, we have created a model in SOLIDWORKS making it as closer to reality as possible.

To minimize the error of the estimate due to the imprecision of the software, we have represented every component of the attitude module, from the external panel to each screw, nut and washer. Subsequently, all the components have been weighted to determine their range of density, due to the uncertainty in the volume and in the measured mass. The mean density of every element was then applied to the correspondent part in the SOLIDWORKS model.

The created model of the SPARTANS spacecraft simulator is shown in Fig. 3.8 and Fig. 3.9.

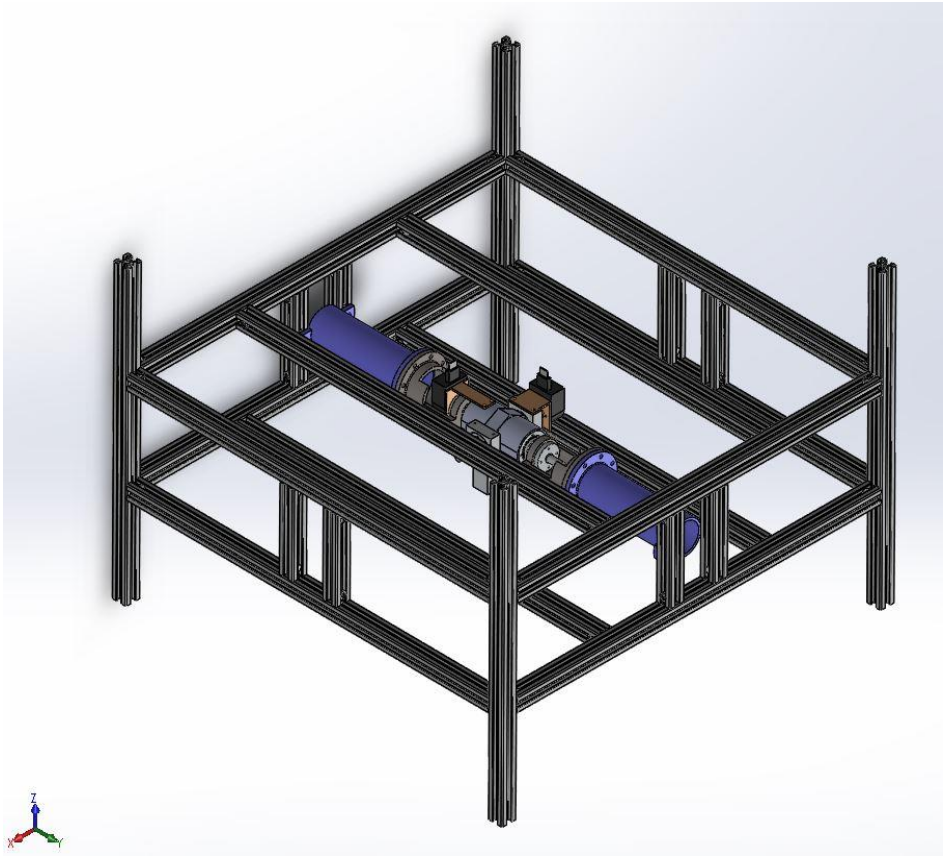

Figure 3.8: Frame of the SPARTANS spacecraft simulator with the mechanical joint that provides the three rotational degrees of freedom.

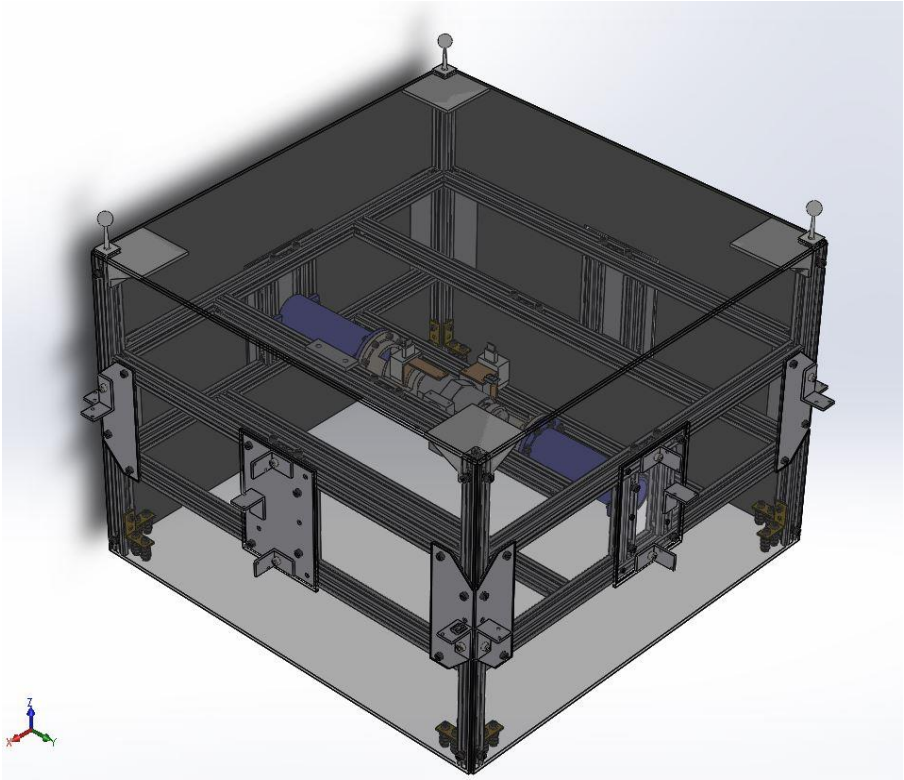

Figure 3.9: Complete model of the SPARTANS spacecraft simulator with all the components.

After the estimation and the application of the density to each component of the SPARTANS spacecraft simulator model, it was possible to evaluate of the mass properties of the attitude module from SOLIDWORKS, which are listed in Table 3.1.

Table 3.1: Evaluated mass properties of the SPARTANS spacecraft simulator model.

Volume		**0.05997 m ^{3}**
Mass		**7.73103 Kg**
Centre of Mass position error	**X**	**0.09 mm**
	Y	**- 0.34 mm**
	Z	**- 0.34 mm**
Principal Moments of Inertia	**I_{XX}**	**0.331655 Kg·m ^{2}**
	I_{YY}	**0.358278 Kg·m ^{2}**
	I_{ZZ}	**0.582930 Kg·m ^{2}**

In the real Attitude Module, its center of mass is almost coincident with the center of rotation of the mechanical joints system. Therefore, the position of the center of mass can be used to obtain a preliminary validation of the results of the Attitude Module mass properties obtained with the SOLIDWORKS model. In that model, indeed, the origin of the reference system is placed at the bottom of the model exactly under the rotation center, so that the nominal positions of the rotation center are $x_{rc} = 0\ mm$, $y_{rc} = 0\ mm$ and $z_{rc} = 280\ mm$. The coordinate of the rotation center obtained with the CAD model are $x_{rc,CAD} = 0.09\ mm$, $y_{rc,CAD} = 0.34\ mm$ and $z_{rc,CAD} = 279.66\ mm$, with an error on the three axes of $0.09\ mm$, $0.34\ mm$ and $0.34\ mm$, respectively.

For a further and more accurate validation, and eventually a correction, of these results it would be possible to determine the actual moments of inertia of the attitude module by carrying out an appropriate experiment using the Trifilar Pendulum.

Later the values of these moments of inertia were varied to evaluate the influence of errors in the knowledge of these parameters on the performances of the navigation filters.

3.4 Measurements' variances

The Kalman filters compute and propagate the state vector which includes all the parameters of the relative dynamics: position, translational velocity, attitude and angular velocity. For an accurate estimation and for evaluating the reliability of the received measurements, the filters need to be tuned using the variances of every state parameter.
For the estimation of all the variances required, we set up Montecarlo analyses starting from the uncertainty of the measurement systems, applying all the transformations and the correction needed, to finally obtain the uncertainties and the variances of all the parameters of the relative dynamics between the target body reference frame and the chaser body reference frame.

The Motion Capture system has an uncertainty of half a millimeter for the position of the spherical marker, which leads to the following uncertainties: around 0.15° for the relative attitude; around a millimeter for the position of the center of mass of the module; three millimeters per second for the translational velocity; around 0.4° per second for the angular velocity (see Table 3.2).

Table 3.2: Uncertainties and variances of the relative dynamics parameters obtained starting from the MC system's uncertainty.

MOTION CAPTURE		**Uncertainty**	**Variance**
Position of the center of mass	**X**	**± 1.19 mm**	**$3.3308 \cdot 10^{-4}$**
	Y	**± 1.51 mm**	**$4.4883 \cdot 10^{-4}$**
	Z	**± 0.93 mm**	**$2.4105 \cdot 10^{-4}$**
Velocity of the center of mass	**V_X**	**± 0.0026 m/s**	**$6.6889 \cdot 10^{-4}$**
	V_Y	**± 0.0028 m/s**	**$8.6918 \cdot 10^{-4}$**
	V_Z	**± 0.0016 m/s**	**$4.0285 \cdot 10^{-4}$**
Attitude expressed with Euler angle	**ϕ**	**± 0.165 °**	**$8.790 \cdot 10^{-4}$**
	θ	**± 0.138 °**	**$8.527 \cdot 10^{-4}$**
	ψ	**± 0.158 °**	**$8.888 \cdot 10^{-4}$**
Attitude expressed with Quaternion	**q_0**	**± 0.00134**	**$3.699 \cdot 10^{-4}$**
	q_1	**± 0.0015**	**$4.872 \cdot 10^{-4}$**
	q_2	**± 0.00127**	**$3.534 \cdot 10^{-4}$**
	q_3	**± 0.0011**	**$3.006 \cdot 10^{-4}$**
Attitude angular velocity	**ω_X**	**± 0.3787 °/s**	**$1.707 \cdot 10^{-4}$**
	ω_Y	**± 0.3713 °/s**	**$1.750 \cdot 10^{-4}$**
	ω_Z	**± 0.3644 °/s**	**$1.767 \cdot 10^{-3}$**

The measurements coming from the monocular camera are affected by various sources of uncertainty. The most important ones are:

- The actual three-dimensional positions of the square ARUCO markers placed on the external surface of the target with respect to their reference frame;
- The two-dimensional positions of the corners of the square markers projected on the image plane;
- The actual position and orientation of the camera optical frame with respect to the reference frame of the inspector;
- The actual position and orientation of the target body reference frame with respect to the square marker reference frame;
- The monocular camera's intrinsic parameters.

Once the monocular camera and the roto-translational matrices, $\frac{I}{C}\boldsymbol{T}$, $\frac{S}{B}\boldsymbol{T}$, have been calibrated, it is possible to estimate the position and the attitude of the target with respect to the chaser starting from the images collected by the monocular camera. These quantities are associated to an uncertainty of around 10 mm for the position and of almost 1° for the attitude. Starting from these uncertainties, another Montecarlo analysis, set up as before, leads to the variances of all the parameters, referred to the measurements derived from the images taken with the monocular camera. The values obtained from this analysis are shown in Table 3.3.

Table 3.3: Uncertainties and variances of the relative dynamics parameters obtained starting from the monocular camera measurements uncertainties.

Monocular camera		**Uncertainty**	**Variance**
Position of the center of mass	**X**	**± 12.2 mm**	**$7.560 \cdot 10^{-3}$**
	Y	**± 8.9 mm**	**$5.485 \cdot 10^{-3}$**
	Z	**± 4.86 mm**	**$2.979 \cdot 10^{-3}$**
Velocity of the center of mass	**V_X**	**± 0.0816 m/s**	**$4.252 \cdot 10^{-2}$**
	V_Y	**± 0.0633 m/s**	**$3.108 \cdot 10^{-2}$**
	V_Z	**± 0.0309 m/s**	**$1.702 \cdot 10^{-2}$**
Attitude expressed with Euler angles	**ϕ**	**± 0.9755 °**	**$5.947 \cdot 10^{-1}$**
	θ	**± 0.7805 °**	**$4.800 \cdot 10^{-1}$**
	ψ	**± 0.4501 °**	**$2.764 \cdot 10^{-1}$**
Attitude expressed with Quaternions	**q_0**	**± 0.0106**	**$4.252 \cdot 10^{-3}$**
	q_1	**± 0.00598**	**$2.668 \cdot 10^{-3}$**
	q_2	**± 0.00773**	**$3.094 \cdot 10^{-3}$**
	q_3	**± 0.00803**	**$4.200 \cdot 10^{-3}$**
Attitude angular velocity	**ω_X**	**± 6.73 °/s**	**3.506**
	ω_Y	**± 5.32 °/s**	**2.929**
	ω_Z	**± 2.79 °/s**	**1.609**

Chapter 4

Kalman filters

Completed the preliminary operations described in the previous chapters, it is possible to set-up the Kalman filters for estimating the relative dynamics between the two satellites. Before starting with the set up, it is important to understand how each filter works and which its functional equations are.

All the Kalman filters formulations descent from the first linear Kalman filter and so they all share the same structure, but naturally each one has its individual characteristics, as described in detail in the following sections.

4.1 Linear Kalman Filter

The linear Kalman Filter has the purpose of providing a state optimal estimate of a discrete-time linear system. Essentially it is a predictor-corrector algorithm based on two principal steps: the prediction or time update step and the measurements update step. These steps can be described by their core equations:

Prediction step: $$\hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-} = \boldsymbol{A_k}\,\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{+} + \boldsymbol{B_k}\,\boldsymbol{u_k}$$

Measurements update step: $$\hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{+} = \hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-} + \boldsymbol{K_{k+1}}\,(\hat{\boldsymbol{z}}_{\boldsymbol{k+1}} - \hat{\boldsymbol{z}}_{\boldsymbol{k+1}}^{-})$$

where: $$\hat{\boldsymbol{z}}_{\boldsymbol{k+1}}^{-} = \boldsymbol{H_{k+1}}\,\hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-}$$

The variables used in the previous equations are described in the following.

- $\hat{\boldsymbol{x}}$ is the state vector; this vector collects all the parameters that has to be estimated that, for the case of this thesis, are the relative position and translation velocity, for the translational analyses, and the relative attitude and angular velocity, for the rotational analyses.

- $\hat{\boldsymbol{z}}$ is the measurements vector, i.e. all the measurements coming from the sensors. For the cases of this thesis, the sensors are the cameras that provide measurements of the relative position and attitude between the satellites. It is important to note that, while the vector $\hat{\boldsymbol{z}}$ comes from the actual measurements of the sensors, $\hat{\boldsymbol{z}}^{-}$ is the measurements vector estimated starting from the state vector $\hat{\boldsymbol{x}}$.

- ***u*** is the input or control vector; this vector collects the parameters that control or influence the propagation of the state vector $\hat{\boldsymbol{x}}$, hence, in this case, the external acceleration acting on the simulators.

- ***A*** is the process matrix, the state transition matrix of the system. With this matrix it is possible to apply the operations needed to propagate the state vector by a time instant. Later the propagations' equation and the relative form of this matrix will be displayed.

- ***B*** is the control matrix; this is needed to apply correctly the components of the control vector ***u*** to the propagation of the state vector.

- ***K*** is the Kalman gain or blending factor; this is necessary to determine how much trust give the two measurements vectors, $\hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{z}}^{-}$, in order to quantifies the influence of the measurement update to the state vector $\hat{\boldsymbol{x}}$.

- ***H*** is the measurement or observation matrix; this matrix transforms the values of the state vector to the ones of the measurements vector. Thanks to this it is possible to compare the values obtained with the propagation with those supplied by the sensors' measurements.

- The index ***k*** stands for the time instant.

- The apex ***+*** and ***–*** stands respectively for the actual value, or *a posteriori*, corrected by the measurements, and the estimated value, or *a priori*, calculated during the prediction.

In the next equations, that complete the algorithm of the filter, there are also the following three matrices.

- ***P***: the matrix of the variances of the state vector's errors. This is necessary for the calculation of the Kalman gain ***K*** as this will influence the trust to put on the estimated measurements vector, $\hat{\boldsymbol{z}}^{-}$.

- ***R***: the matrix of the variances of the measurements vector's errors. This also is needed in the calculation of the Kalman gain ***K*** but this will influence the trust to put on the actual measurements vector $\hat{\boldsymbol{z}}$.

- ***Q***: the matrix of the process noises' variances. This is needed in the propagation of the matrix ***P*** in order to take into accounts also the process noises within the new state vector's errors variances.

The prediction step is responsible for the propagation of the state vector and of its covariances matrix forward in time, leading to an *a priori* estimate of these quantities at the next time instant (**k+1**).
In the update step there is the input to the filter of the new measurements vector, collected at a certain time instant. Employing this measurements vector, the correction and update of the state vector and its covariances matrix are carried out obtaining an improved *a posteriori* estimate for the state parameters at the time instant of the input measurements vector.

The linear Kalman filter algorithm is shown in Fig. 4.1.

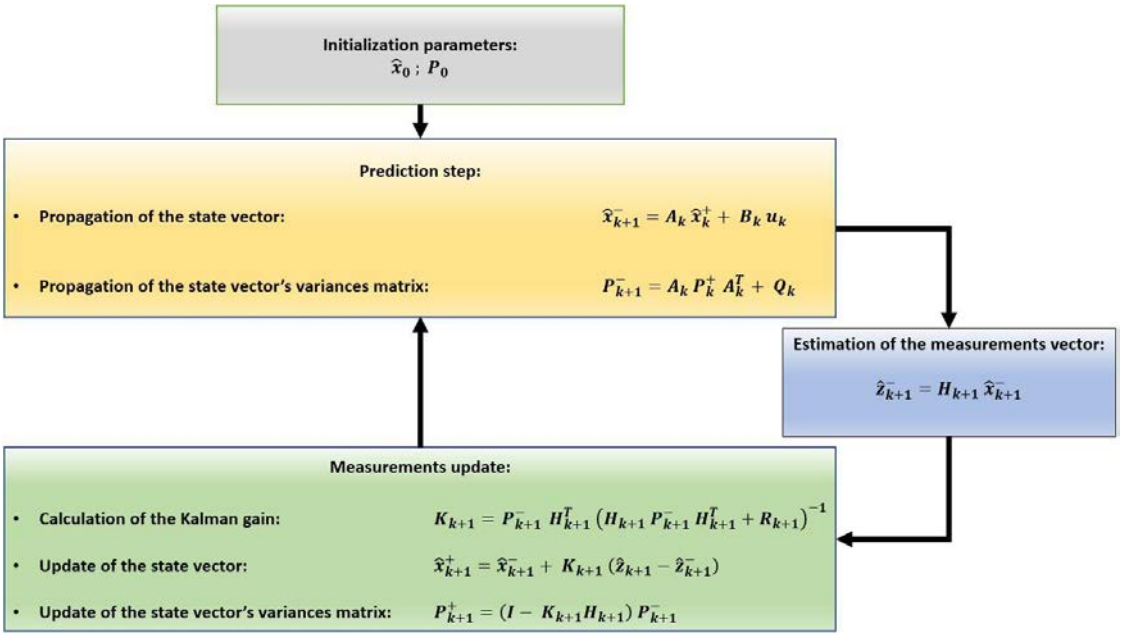

Figure 4.1: Outline of the linear Kalman Filter.

At the beginning of its process, the linear Kalman filter needs an initialization phase in order to start the first prediction step, after this, the values calculated at each step are used for all the other following operation proceeding with the next steps. However, the control vector ***u*** has to be always supplied to the filter at every time-instant because it is associated to external actions that cannot be predicted by the filters. This is clear for the case of this thesis, where the control vector represents the effects of the external forces applied to the spacecraft for some maneuvers or as disturbances.

For every time-instant the filter gives as output the state vector, that is composed by the parameters of interest, the relative pose and dynamics for the current case. So, it can operate continuously as well as it is supplied by at least the control vector and by the measurements vector for a higher precision. This nature of the Kalman filter makes its practical implementations much more feasible than other estimation filters.

The key equation that determine the precision of this filter is the Kalman gain calculation:

$$\boldsymbol{K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1} = \frac{P_k^- H_k^T}{\left(H_k P_k^- H_k^T + R_k\right)}}$$

This equation is designed to provide to the Kalman gain the value that minimizes the *a posteriori* error covariance matrix $\boldsymbol{P^+}$. From this equation it is easy to see the influences of the covariance matrices ***P*** and ***R*** on the value of the Kalman gain. In fact, if the matrix ***R*** of the measurements error variances approaches a zeros matrix, the Kalman gain appears to tend towards the values of $\boldsymbol{H_k^{-1}}$. On the other side, if the *a priori* estimated matrix of the state vector's variances $\boldsymbol{P^-}$ approaches a zeros matrix, the Kalman gain approaches a zeros matrix too.

From the measurement update equation, it is possible to evaluate the implications of these variation:

$$\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{+} = \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-} + \boldsymbol{K}_{\boldsymbol{k}} \left(\hat{\boldsymbol{z}}_{\boldsymbol{k}} - \boldsymbol{H}_{\boldsymbol{k}} \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-}\right)$$

Here it is clear that if the Kalman gain is equal to $\boldsymbol{H}_{\boldsymbol{k}}^{-\mathbf{1}}$ this equation simply becomes:

$$\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{+} = \boldsymbol{H}_{\boldsymbol{k}}^{-\mathbf{1}} \hat{\boldsymbol{z}}_{\boldsymbol{k}}$$

In this way, during the update, the estimated *a priori* state vector is worthless, because the calculation of the *a posteriori* state vector is based only on the measurement vector relaying all the trust to the measurements collected by the sensors.

On the other side, if the Kalman gain $\boldsymbol{K}_{\boldsymbol{k}}$ is a zeros matrix, the equation results:

$$\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{+} = \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-}$$

In this case, the measurements vector and, in general, the sensors are useless, because there is not any calculation of the *a posteriori* state vector hence it is equal to the *a priori* one, so only the prediction step has to be carried out, while the update step can be skipped.

These are the extreme values for the Kalman gain and they are helpful for the understanding of its influence on the updating, as described below.

- If the matrix ***R*** tends to be close to zero, the Kalman gain will increase tending to $\boldsymbol{H}_{\boldsymbol{k}}^{-\mathbf{1}}$ and so the residuals $\left(\hat{\boldsymbol{z}}_{\boldsymbol{k}} - \boldsymbol{H}_{\boldsymbol{k}} \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-}\right)$ will be weighted more heavily up to the limit:

  $$\lim_{\boldsymbol{R}_{\boldsymbol{k}} \to \mathbf{0}} \boldsymbol{K}_{\boldsymbol{k}} \to \boldsymbol{H}_{\boldsymbol{k}}^{-\mathbf{1}} \Rightarrow \hat{\boldsymbol{x}}_{\boldsymbol{k}}^{+} \to \boldsymbol{H}_{\boldsymbol{k}}^{-\mathbf{1}} \hat{\boldsymbol{z}}_{\boldsymbol{k}}$$

  This means that while the matrix ***R*** decreases, the actual measurements vector is trusted more and more, whereas the estimated measurements vector is less and less trusted. All this has its reasons, because if the matrix ***R*** is low, it means that the errors of the sensors are low and so the measurements taken are very accurate. Given this, the measurements vector can be highly trusted. In the limit, if the matrix ***R*** is zero, it means that the measurements are perfect, without any errors, and for this it is better to lay only on the measurements for pursuing the update step.

- If the matrix $\boldsymbol{P^-}$ tends to be close to zero, the Kalman gain will also decrease and tend to zero and so the residuals $(\hat{\boldsymbol{z}}_{\boldsymbol{k}} - \boldsymbol{H_k}\,\hat{\boldsymbol{x}}_{\boldsymbol{k}}^-)$ will be weighted less and less heavily down to the limit:

$$\lim_{P_k^- \to 0} K_k \to 0 \;\Rightarrow\; \hat{x}_k^+ \to \hat{x}_k^-$$

  This means that while the matrix $\boldsymbol{P^-}$ decreases, it is the estimated measurements vector the one that is more and more trusted whereas the actual measurements vector is less and less trusted. In this case, if the matrix $\boldsymbol{P^-}$ is low, it means that the errors associated to the state vector and to the process are low and for this reason the estimated *a priori* state vector appears to be very accurate, so it can be highly trusted. In the limit, if the matrix $\boldsymbol{P^-}$ is zero, it means that the estimation process and the *a priori* state vector are perfect and without any errors or noises, so it is more convenient to skip the update step and rely only on the prediction.

  Even though the Unscented version of the Kalman filter uses a different equation to calculate the Kalman gain, all these considerations, on the influence of the matrices $\boldsymbol{P}$ and $\boldsymbol{R}$, are proper also in that case.

The linear Kalman filter cannot be used with non-linear systems, because its direct application will be very long and difficult and it will not be able to provide the optimal state estimation. With this filter it is possible to study only linear system, because it uses always the same **A** and **B** matrices defined before the start of the filtering process. This allows the analysis to be very fast, but, without a continuous recalculation of the process and the control matrices, it leads to the impossibility to analyze a non-linear system. To overcome this limitation, the Extended Kalman filter has been developed.

Later in this thesis, when this filter was applied to the filtering of the measurement taken with the monocular camera, a little non-linearity was introduced, given by the propagation time that changes following the capture-time of the images. Rigorously speaking, for that case the Kalman filter becomes an Extended Kalman filter, but the non-linearity introduced is very small, so in this thesis it will still be considered as a linear Kalman filter.

4.3 Extended Kalman filter

The Extended Kalman filter is the first variant of the Kalman filter and it was created with the aim of applying this kind of filter also to non-linear systems.

In order to achieve this purpose, it does not present many differences with respect to the classic filter, but an approximation is required. In fact, the application of the Extended Kalman filter demands the linearization of the system to be analyzed. Consequently, following this approximation, a non-linear system has to be linearized in many discrete linear portions instant by instant. In this way, it is possible to apply the linear Kalman filter singularly to each of these portions. To study the whole linearized system with only one filtering procedure there is just the need to recalculate for every step the two matrices ***A*** and ***B***, respectively the process and the control matrices. This is how the Extended Kalman filter works and that is why the form and the equations of this filter are exactly the same of the classic Kalman filter, with the exception of the recalculation of the ***A*** and ***B*** matrices, even though they are designed for different kind of systems.

However, there are some important considerations that must be considered.

First, the Extended Kalman filter requires the continuous recalculation of the two propagation matrices and this causes a slight reduction of the speed of the filter, making this a little slower than the classic linear Kalman filter.

Second, since this filter just represents the application of a linear filter to an approximated linearized system, it leads to a suboptimal solution of the non-linear estimation problem and also it does not grant the stability of the solution: it can work very well for a long period and then suddenly diverge irremediably.

Another important issue of the Extended Kalman filter to consider is the fact that the linearization approximation has to be carried out very carefully because it can lead to large error or to loss of time in the following filtering procedure. This depends on the non-linearity of the system to analyze.

- If the non-linearity of the system is high, the linearization has to be very refined, with very small discrete linear portions. Using a mild linearization in the filtering procedure with large discrete portions, it will not be possible to follow correctly the non-linear evolution of the system and this will cause the growth of large errors in the final results.
- If the non-linearity of the system is quite low, it is inconvenient to use a refined linearization with a great amount of very small portions, because during the filtering procedure every step implies different calculations with their relative computational time. In the end this can cause the analysis to be much more time-consuming without obtaining an actual advantage in the precision of the results.

This consideration is an issue for the analyses of non-linear systems; therefore, it involves also the Unscented Kalman filters when they are used for the filtering of this kind of systems.

Later in this thesis, two different linearization techniques were adopted to study the relative attitude motion, which is quite non-linear. One linearization technique, applied to the cases that use the measurement from the Motion Capture system, is characterized by a frequency of 500 Hz for the discretization; the other linearization method, applied to the cases that use the measurement from the monocular camera, is characterized by a frequency of 100 Hz.
In the final results it was possible to verify that the first approximation is excessively refined, while the second one is much more suitable. Indeed, the precision of the results is not affected by the different linearization methods and analysis frequency, while the computational time shows great influence to it, becoming much higher in the first cases.

The algorithm of the Extended Kalman filter is shown in Fig. 4.2.

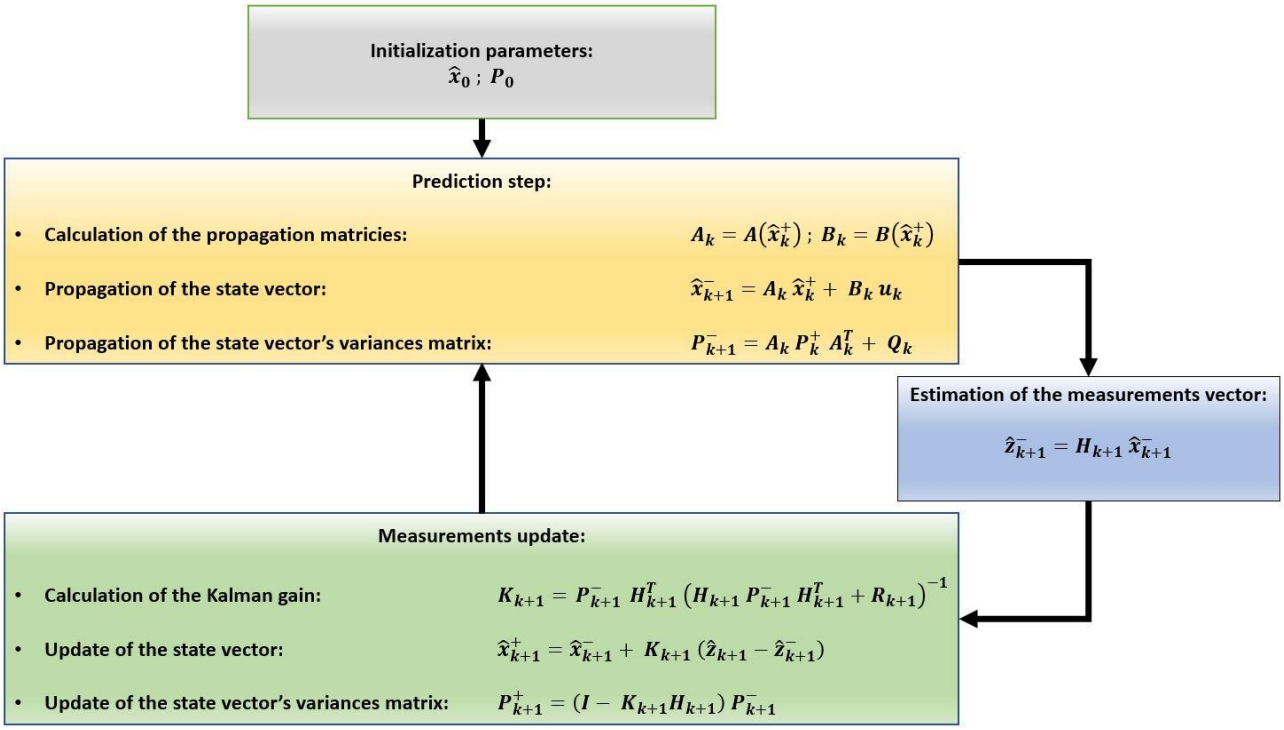

Figure 4.2: Outline of the Extended Kalman Filter.

As already mentioned, the equations and algorithm of this filter are the same of the linear one except for the introduction of the recalculation of the two propagation matrices ***A*** and ***B*** at each prediction step, and therefore all the considerations seen above are suitable also for this case.

4.4 Unscented Kalman Filter

A further evolution to the Kalman filter is brought by the Unscented family of this filter. This new family was born with the aim of increasing the accuracy of the Kalman filters when dealing with non-linear systems. They, indeed, ensure an accuracy of at least the second-order Taylor series approximation without involving too complex procedures and calculations.

This result is obtained thanks to a new approach known as the Unscented Transformation (UT). This is the main difference between the basic Kalman filter and the Unscented Kalman filter. The Unscented Transformation is a new method of predicting the mean state vector and the error variances that is based on deterministic sampling. This involves only the propagation step of the filter and not the measurement update step, which is therefore carried out in the same way.

The difference between the classic and the new approach is easy to understand looking at the two pictures below.

The propagation step of the classic Kalman filter, which includes also the Extended one, can be seen in Fig. 4.3.

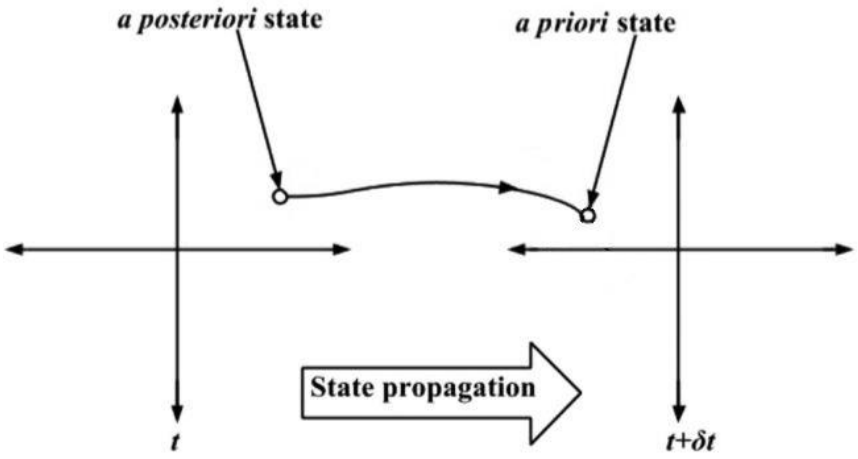

Figure 4.3: State propagation strategy for the standard Kalman filters [11].

Here it is possible to see that, starting from one *a posteriori* state vector, the propagation takes place to find one estimated *a priori* state vector at the next time instant.

The propagation step of the Unscented Kalman filter is shown in Fig. 4.4.

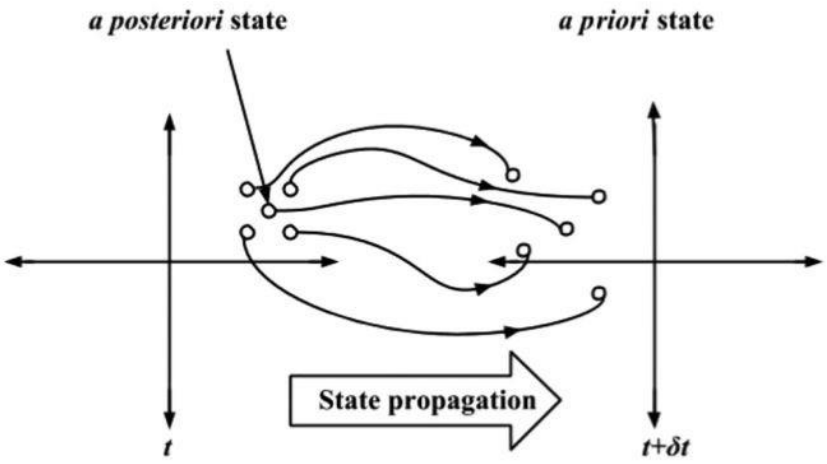

Figure 4.4: State propagation strategy for the Unscented Kalman Filter [11].

Here there is not only one state vector, but there is a cloud of starting points that are propagated into an estimated cloud of points at the next time instant; this is thanks to the Unscented Transformation.

For a system where the state vector has ***n*** state elements, with the UT there is the creation of ***2n + 1*** sigma points starting from the *a posteriori* state vector.
These sigma points are new state vectors that approximate the original *a posteriori* state vector; in fact, they are created starting from this vector and then adding, in a suitable way, the variances associated to the state vector itself. The sigma points are realized such as their mean is equal to the starting state vector and also their variances are equal to the ones of the starting state vector.
All these sigma points form the cloud of state vector points that are propagated into the estimated cloud of *a priori* state vector points. Then, to obtain the final estimated *a priori* state vector, these propagated sigma points are mediated with weights. For the calculation of the variances' matrices and for the Kalman gain there are new suitable equations that use the information from every sigma point.

All these operations have to be carried out at every time instant to ensure the correct propagation of the state vector. It is easy to realize that this new method implies an increase of the computational time needed for the filtering procedure. Beyond the creation of the sigma points, there is also their propagation and later use for the determination of the various matrices and this requires more time because, for every step, the operations that have to be performed are multiplied by ***2n + 1***.

In addition, to ensure higher accuracy to the Unscented family of the Kalman filter, the state vector is augmented by the relative error of each element of the state vector. In this way, the parameter ***n*** and the number of sigma points are greater, and the computational time increases.

For the case of this thesis, the following parameters were adopted.

- For the translational analysis, the state vector elements are the three position coordinates and the three translational velocities of the center of mass. So, the augmented state vector has twelve elements and the sigma points are **25**.
- For the attitude analysis, the state vector elements are the four-element quaternion and the three attitude angular velocities. So, the elements of the augmented state vector are fourteen and the sigma points are **29**.

The equations for the calculation of the sigma points and their respective weights are:

- Definition of the sigma points:

$$\boldsymbol{Y_i(k)} = \begin{cases} \hat{\boldsymbol{x}}^{+}(\boldsymbol{k}) & (\boldsymbol{i = 0}) \\ \hat{\boldsymbol{x}}^{+}(\boldsymbol{k}) + \boldsymbol{\Delta Y_i} & (\boldsymbol{i = 1, 2, \dots 2n}) \end{cases}$$

$$\boldsymbol{\Delta\, Y_i} = \begin{cases} \sqrt{(\boldsymbol{n+\xi})\, \boldsymbol{P^{+}_{xx}(k)}_{\,i}} & (\boldsymbol{i = 1,\ 2,\ \dots n}) \\ -\sqrt{(\boldsymbol{n+\xi})\, \boldsymbol{P^{+}_{xx}(k)}_{\,i}} & (\boldsymbol{i = n+1,\ n+2, \dots 2n}) \end{cases}$$

- Definition of the weights:

$$\boldsymbol{W_i} = \begin{cases} \dfrac{\boldsymbol{\xi}}{\boldsymbol{n+\xi}} & (\boldsymbol{i = 0}) \\ \dfrac{\boldsymbol{1}}{\boldsymbol{2(n+\xi)}} & (\boldsymbol{i = 1, 2, \dots 2n}) \end{cases}$$

Here it is possible to see that the creation of the sigma points $\boldsymbol{Y_i(k)}$ starts from the *a posteriori* state vector and then the parameters $\boldsymbol{\Delta\, Y_i}$ are added. Each of these parameters is a column vector defined starting from the columns of the matrix $\boldsymbol{P^{+}_{xx}}$.
In these equations the factor $\boldsymbol{P^{+}_{xx}(k)}_{\,i}$ stands for the *i-th* column of the matrix $\boldsymbol{P^{+}_{xx}}$.

The constant $\boldsymbol{n}$ is the number of elements of the augmented state vector, as seen before, and the constant $\boldsymbol{\xi}$ is a scaling factor that can vary between $\boldsymbol{0}$ and $\boldsymbol{3 - n}$. It is important to notice that when this factor is set to zero, also the first weight equals zero and this implies that the *a posteriori* state vector will not be taken into account for the following operations. For the cases studied in this thesis, the variation of this parameter does not influence any results in almost every analysis except for the one that will be discussed later.

The sigma points and their weights are defined in this ways in order to have the weighted mean of the sigma points equal to the starting state vector $\hat{\boldsymbol{x}}^{+}(\boldsymbol{k})$ and the variances of the sigma points equal to the matrix $\boldsymbol{P^{+}_{xx}(k)}$ of the variances of the state vector $\hat{\boldsymbol{x}}^{+}(\boldsymbol{k})$.

Knowing how the Unscented Transformation works it is possible to analyze the whole algorithm of the Unscented Kalman filter, shown in Fig. 4.5.

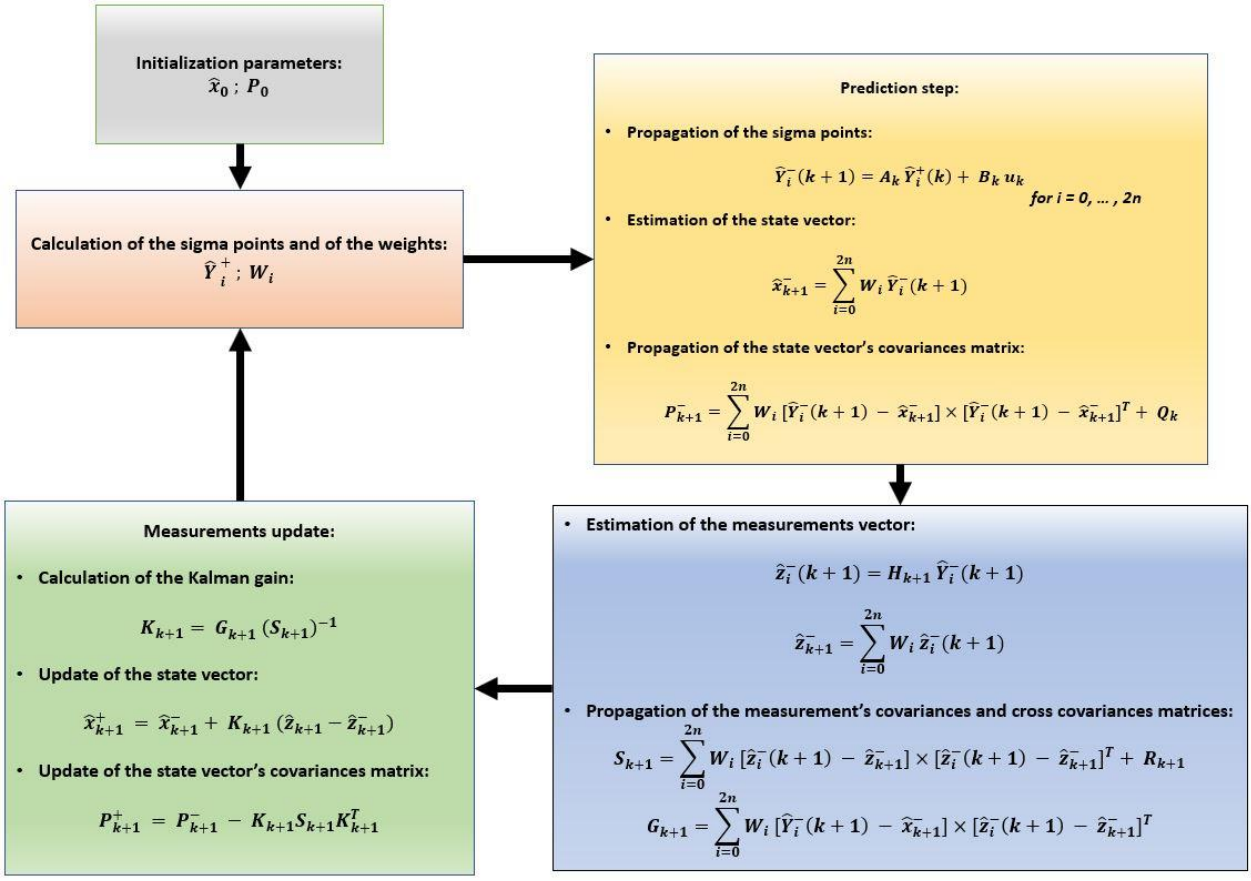

Figure 4.5: Outline of the Unscented Kalman Filter.

Following the creation of the sigma points, each one of these new vectors is propagated at the next time instant, finding the *a priori* sigma points $\hat{Y}_i^-$. To obtain the *a priori* state vector, $\hat{x}_{k+1}^-$, a weighted mean of these estimated points has to be done. Then using this mean value and each of the new sigma points it is possible to directly calculate the actual covariances matrix of the new sigma points that is the *a priori* covariances matrix of the state vector, P_{k+1}^-, adding at the end the process noise variances' matrix Q_k.

From each propagated sigma points there is also the estimation of ***2n + 1*** measurements vector, $\hat{z}_i^-$, that will be mediated with their respective weights in order to find the *a priori* estimated measurement vector, $\hat{z}_{k+1}^-$.

As done for the state vector, it is possible to calculate directly the covariances matrix of the measurement vector using the *a priori* estimated one and the cloud of the ***2n + 1*** measurement points, obtaining the new matrix S_{k+1}, adding at the end the measurements noise variances' matrix R_{k+1}. Then, using the two clouds of the estimated vectors, all the $\hat{Y}_i^-$ and the $\hat{z}_i^-$, and the two *a priori* vector, the state and the measurement ones, $\hat{x}_{k+1}^-$ and $\hat{z}_{k+1}^-$, it has to be calculated the cross covariances matrix between the two sets of points, obtaining the matrix G_{k+1} .

These two last matrices are essential for the calculation of the Kalman gain for this kind of filter, which is done using the equation:

$$\boldsymbol{K_{k+1} = G_{k+1}\,(S_{k+1})^{-1}}$$

This equation is different from the one seen and analyzed for the two first Kalman filters, but the consideration done for that formulation on the influence of the two matrices $\boldsymbol{P_{k+1}^{-}}$ and $\boldsymbol{R_{k+1}}$ are still correct in this case, only that here they are not so evident.

At the end, using the just calculated Kalman gain there is the measurement update phase where the *a posteriori* state vector and the *a posteriori* state vector covariances matrix, $\boldsymbol{\hat{x}_{k+1}^{+}}$ and $\boldsymbol{P_{k+1}^{+}}$, are calculated.

Clearly, in addition to all these equations, if this filter is used for non-linear systems the two propagation matrices, ***A(k)*** and ***B(k)***, has to be recalculated before the prediction for every time instant, as seen for the Extended Kalman filter.

It is easy to understand that the Unscented Kalman filter is more time demanding than the two basic ones, since the Unscented Transform implies the creation of ***2n*** more vectors that have to be used and then some new equations during the whole filtering procedure increase the complexity and time-cost of this filter. However, all this complexity addition ensures more stability and precision to this filter, preventing it to diverge irremediably and leading to a better solution of the non-linear estimation problem, solving the issues of the Extended Kalman filter.

4.5 Single Propagation Unscented Kalman Filter

Most physical systems are described by non-linear continuous-time differential equations and for an accurate state propagation of these highly non-linear systems it is inevitable to perform a numerical integration. In these cases, using an Unscented Kalman filter implies multiple numerical integration, one for every sigma point, at every prediction step to calculate the *a priori* state vector. For this reason, the implementation of the UKF is much more computationally expensive than of the Extended Kalman filter, but the second one can hardly lead to an accurate estimation.

For this reason, there was the need to reduce the computational time of the UKF without losing its accuracy and that is why the Single Propagation Unscented Kalman Filter (SPUKF) and the Extrapolated Single Propagation Unscented Kalman Filter (ESPUKF) have been developed.

At first to reduce the computational time of the UKF the most obvious strategy seemed to be the reduction of the number of sigma points to be created and propagated and from this perspective various solutions came out. However, operating this reduction can lead to a degraded estimation performance and does not ensure a great reduction of time, which is proportional to the number of spared sigma points, while the accuracy of the filtering procedure is inversely proportional to this number.

Later a new approach has been developed; unlike the previous ones, in this new method there is the possibility of reducing the computational time without reducing the number of the sigma points and so without losing the accuracy of the Unscented Kalman filter.
In this new approach, the Unscented Transformation is bypassed, the *a posteriori* state vector is directly propagated to the next time instant and only then the other ***2n*** sigma points are calculated. In this way there is only the propagation of one state vector instead of ***2n + 1*** propagations.
The UKF that takes advantage of this strategy is the Single Propagation Unscented Kalman Filter (SPUKF).

The propagation of the single state vector and the later creation of the sigma points that bypass the standard Unscented Transform can be easily seen from Fig. 4.6.

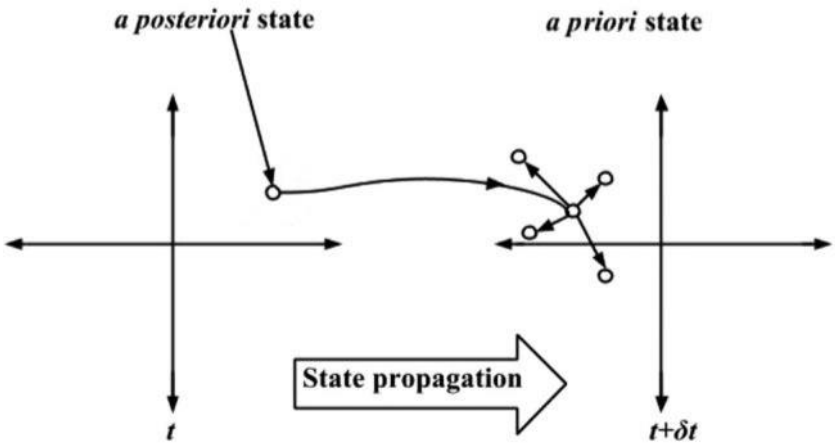

Figure 4.6: State propagation strategy for the Single Propagation Unscented Kalman Filter [11].

This is how this new filter bypass the classic Unscented Transformation. Here it is possible to see that there is only a single propagation, the one of the *a posteriori* state vector, and not ***2n + 1*** as in the UKF. From this it is clear that the more the number of sigma points and the complexity of the equations that describe the non-linear system, the faster this new strategy when compared to the classic UKF.

The key feature of this filter is the calculation of the sigma points after the propagation that is realized using a first-order Taylor series approximation. Certainly, this approximated calculation leads to a lower estimation accuracy for the SPUKF with respect to the UKF, but the error is still very low and comparable to the second-order terms of the Taylor series expansion.

To understand how the new calculation of the sigma points is pursued, the classic creation and propagation of them has to be analyzed and approximated as shown below.

Following the conventional Unscented Transformation, the starting sigma points are:

$$\boldsymbol{Y_i\,(k) = \hat{x}^{+}(k) + \Delta Y_i} \qquad \textbf{\textit{(i = 1, 2, … , 2n)}}$$

where the augmentation factor is:

$$\boldsymbol{\Delta\, Y_i =} \begin{cases} \sqrt{(n+\xi)\, P^{+}_{xx}(k)_{\,i}} & (i = 1,\ 2,\ \ldots n) \\ -\sqrt{(n+\xi)\, P^{+}_{xx}(k)_{\,i}} & (i = n+1,\ n+2, \ldots 2n) \end{cases}$$

After the propagation and using a Taylor series expansion, they become:

$$\boldsymbol{Y_i\,(k+1) \;=\; F\big(\,Y_i\,(k)\,, u(k)\big) \;=\; F\big(\,\hat{x}^{+}(k)\,, u(k)\big) + D_{\Delta Y_i}\, F \;+\; \frac{D^2_{\Delta Y_i}\, F}{2!} + \frac{D^3_{\Delta Y_i}\, F}{3!} + \cdots}$$

Stopping at the first order approximation this turns into:

$$\boldsymbol{Y_i\,(k+1) \;\approx\; F\big(\,\hat{x}^{+}(k)\,, u(k)\big) + D_{\Delta Y_i}\, F}$$

Here $\boldsymbol{F}$ is the total propagation function that define the process matrix $\boldsymbol{A}$ and also the control matrix and vector, $\boldsymbol{B}$ and $\boldsymbol{u}$. In fact:

$$\boldsymbol{F\big(\,\hat{x}^{+}\,(k)\,, u(k)\big) = \; A(k)\,\hat{x}^{+}(k) + B(k)\,u(k) = \;\hat{x}^{-}(k+1)}$$

Then the other parameter of the series expansions is the total differential of the function $\boldsymbol{F}$ and can be expressed as:

$$\boldsymbol{D_{\Delta Y_i}\, F = \frac{\partial F}{\partial x}\Big|_{\hat{x}^+(k)}\, \Delta Y_i}$$

The function $\boldsymbol{F}$ depends from the state vector only for the process matrix $\boldsymbol{A}$ portion and not for the control portion, so the differential can be evaluated as:

$$\boldsymbol{\frac{\partial F}{\partial x} = \; e^{\,J\,dt}}$$

where $\boldsymbol{J}$ is the Jacobian matrix of the process function $\boldsymbol{f}$ evaluated for $\hat{\boldsymbol{x}}^+(\boldsymbol{k})$**.** The function $\boldsymbol{f}$ defines the state vector derivative starting from the state vector itself: $\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\,\hat{\boldsymbol{x}}\,)$. Using this function together with the chosen control and propagation models it is possible to define the total propagation function $\boldsymbol{F}$.

In the end the propagated sigma points can be expressed with the equation:

$$\boldsymbol{Y_i\,(k+1)\;\approx\;\hat{x}^-(k+1) +\; e^{\,J\,dt}\;\Delta Y_i}$$

This is the core equation with which it is possible to create the sigma points after the propagation of the *a posteriori* state vector. So, to bypass the classic Unscented Transformation there is the sole need to calculate the Jacobian matrix $\boldsymbol{J}$ at every time instant and then use it to evaluate the sigma points with their respective augmentation factor $\boldsymbol{\Delta Y_i}$.

With the approximation some little uncertainty can come up, but the expression appears much more simplified and, in this way, the SPUKF can be faster than the UKF without losing too much accuracy.

The whole algorithm of this new filter can be seen in Fig. 4.7.

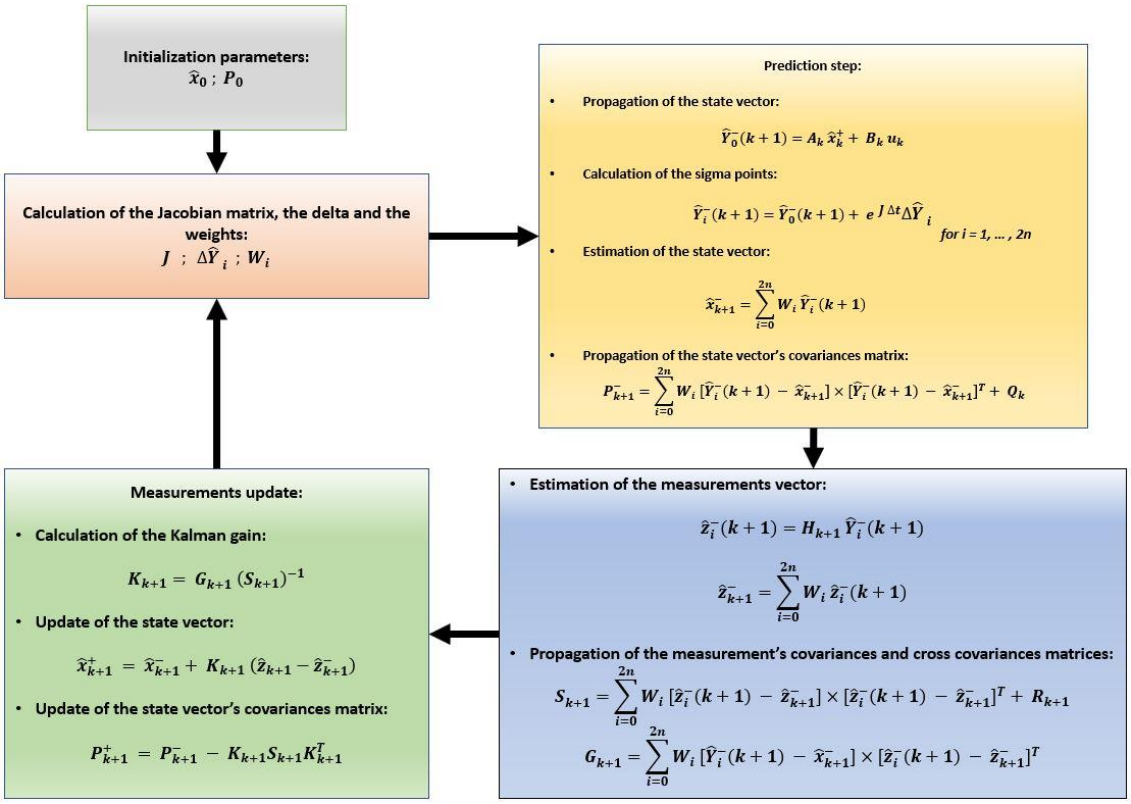

Figure 4.7: Outline of the Single Propagation Unscented Kalman Filter.

Here it is possible to see that after the calculation of the already propagated sigma points, all the other operations, and the correspondent considerations, are the same seen for the basic Unscented Kalman filter.

The differences stand before and during the prediction step. In the SPUKF, before starting the propagation, there is the calculation of the Jacobian matrix, of the augmentation factors and of the weights; they are needed to the later creation of the sigma points. Then, during the prediction phase there is the propagation of the state vector and the following creation of the sigma points. From here on, all the remaining operations concerning the calculation of the covariances matrices, the Kalman gain and the measurement update are exactly the same of the UKF.

4.6 Extrapolated Single Propagation Unscented Kalman Filter

Based on the same strategy of the SPUKF, the Extrapolated Single Propagation Unscented Kalman Filter (ESPUKF) has been developed following the same aim of speeding up the UKF, but also the aim of maintaining the most of its accuracy and so reduce the error on the *a priori* state prediction.

To achieve a better accuracy using the new method for the creation of the sigma points, the second-order Taylor series terms should be included in the *a priori* state vector approximation, but this would involve the calculation of the Hessian matrix of the process function $\boldsymbol{f}$, in addition to its Jacobian matrix, that is very difficult for complex non-linear system with many state elements. To avoid this excessive complexity, this filter still involves the bypass of the Unscented Transformation, for the reduction of the computational time, but then adds to this the Richardson Extrapolation method, because it is known for improving the accuracy of an approximation technique by an order of the Taylor series terms.

So, the UT is still bypassed with the single propagation of the state vector and the following creation of the sigma points, as discussed for the SPUKF (Fig. 4.6). Here, however, the later creation of the sigma points is different and provides better accuracy. In fact, that was the weakness of the precedent filter because it involved an approximation that implies the growth of some uncertainty.

To understand how the Richardson Extrapolation improves the creation of the sigma points, the previous procedure, seen for the SPUKF, has to be changed a little.

As already seen, following the conventional Unscented Transformation, the starting sigma points are:

$$\boldsymbol{Y_i\,(k) = \hat{x}^{+}(k) + \Delta Y_i} \qquad \textbf{\textit{(i = 1, 2, … , 2n)}}$$

where the augmentation factor is:

$$\boldsymbol{\Delta\, Y_i = \begin{cases} \sqrt{(n+\xi)\, P^{+}_{xx}(k)_{\,i}} & (i = 1,\ 2,\ \ldots n) \\ -\sqrt{(n+\xi)\, P^{+}_{xx}(k)_{\,i}} & (i = n+1,\ n+2, \ldots 2n) \end{cases}}$$

For the previous filter, the Taylor series expansion was cut at the first-order terms obtaining in the end the expression:

$$\boldsymbol{Y_i\,(k+1)\;=\;F\big(\,Y_i\,(k)\,,u(k)\big)\;=\;F\big(\,\hat{x}^{+}(k)+\,\Delta Y_i\,,u(k)\big)\;\approx\;F\big(\,\hat{x}^{+}(k)\,,u(k)\big)+\,D_{\Delta Y_i}\,F}$$

which in its explicit form is:

$$\boldsymbol{Y_i\,(k+1)\;\approx\;\hat{x}^{-}(k+1)+\,e^{\,J\,dt}\;\Delta Y_i}$$

This formulation is carried on with the parameter:

$$\boldsymbol{N_1\,(\Delta Y_i)\;=\;F\big(\,\hat{x}^{+}(k)\,,u(k)\big)+\,D_{\Delta Y_i}\,F}$$

Employing the Richardson Extrapolation, the Taylor series expansion is performed in two steps splitting the augmentation factor in two halves:

$$\boldsymbol{Y_i\,(k+1)\;=\;F\big(\,Y_i\,(k)\,,u(k)\big)\;=\;F\left(\,\hat{x}^{+}(k)\;+\;\frac{\Delta Y_i}{2}\;+\;\frac{\Delta Y_i}{2}\,,u(k)\right)}$$

The Taylor series expansion is applied two times, one for each $\boldsymbol{\frac{\Delta Y_i}{2}}$, stopping in both cases at the first terms. The first step is:

$$\boldsymbol{Y_i\,(k+1)\;\approx\;F\left(\,\hat{x}^{+}(k)\;+\;\frac{\Delta Y_i}{2}\;\,,u(k)\right)+\,D_{\frac{\Delta Y_i}{2}}\,F\,\big|_{\hat{x}^{+}(k)\,+\,\frac{\Delta Y_i}{2}}}$$

The second step is:

$$\boldsymbol{Y_i\,(k+1)\;\approx\;F(\,\hat{x}^{+}(k)\,)+\,D_{\frac{\Delta Y_i}{2}}\,F\,\big|_{\hat{x}^{+}(k)}\;+\;D_{\frac{\Delta Y_i}{2}}\,F\,\big|_{\hat{x}^{+}(k)\,+\,\frac{\Delta Y_i}{2}}}$$

This final expression is condensed in the parameter:

$$\boldsymbol{N_2\left(\frac{\Delta Y_i}{2}\right)\;=\;F(\,\hat{x}^{+}(k)\,)+\,D_{\frac{\Delta Y_i}{2}}\,F\,\big|_{\hat{x}^{+}(k)}\;+\;D_{\frac{\Delta Y_i}{2}}\,F\,\big|_{\hat{x}^{+}(k)\,+\,\frac{\Delta Y_i}{2}}}$$

Considering the approximation:

$$\boldsymbol{F}\left(\,\hat{\boldsymbol{x}}^{+}(\boldsymbol{k})\,,\boldsymbol{u}(\boldsymbol{k})\right)\;\approx\;\boldsymbol{F}\left(\;\hat{\boldsymbol{x}}^{+}(\boldsymbol{k})\;+\;\frac{\boldsymbol{\Delta Y_i}}{\mathbf{2}}\;\,,\boldsymbol{u}(\boldsymbol{k})\right)$$

the equation for the calculation of the propagated sigma points become:

$$\boldsymbol{Y_i}\;(\boldsymbol{k}+\mathbf{1})=\mathbf{2}\boldsymbol{N_2}\;\left(\frac{\boldsymbol{\Delta Y_i}}{\mathbf{2}}\right)-\;\boldsymbol{N_1}\;(\boldsymbol{\Delta Y_i})$$

In the end, exploiting some simplification, this equation in its explicit form is:

$$\boldsymbol{Y_i}\;(\boldsymbol{k}+\mathbf{1})=\;\hat{\boldsymbol{x}}^{-}(\boldsymbol{k}+\mathbf{1})+\;\boldsymbol{e}^{\,\boldsymbol{J_2}\,\boldsymbol{dt}}\;\boldsymbol{\Delta Y_i}$$

where:

$$\boldsymbol{e}^{\,\boldsymbol{J_2}\,\boldsymbol{dt}}=\left.\frac{\boldsymbol{\partial F}}{\boldsymbol{\partial x}}\right|_{\hat{x}^{+}(k)\,+\,\frac{\Delta Y_i}{2}}\;=\;\left.\boldsymbol{D}_{\boldsymbol{\Delta Y_i}}\,\boldsymbol{F}\right|_{\hat{x}^{+}(k)\,+\,\frac{\Delta Y_i}{2}}$$

Here $\boldsymbol{J_2}$ stands for the Jacobian matrix of the process function $\boldsymbol{f}$, evaluated for each sigma point using its related semi-augmented vector $\hat{\boldsymbol{x}}^{+}(\boldsymbol{k})\;+\;\frac{\boldsymbol{\Delta Y_i}}{\mathbf{2}}$.

So, it is possible to see that, at the end, the changes between the SPUKF and the ESPUKF strategies rely only on the Jacobian matrix. In the first case, that matrix was only one for all the sigma points, in this case every sigma point has its own Jacobian matrix calculated using their semi-augmented vector, the $\hat{\boldsymbol{x}}^{+}(\boldsymbol{k})\;+\;\frac{\boldsymbol{\Delta Y_i}}{\mathbf{2}}$.

Clearly, this new method implies more computational time because it involves the calculation of ***2n*** Jacobian matrices at every time instant, but in this way, it leads to better accuracy.

The whole algorithm of the ESPUKF is shown in Fig. 4.8.

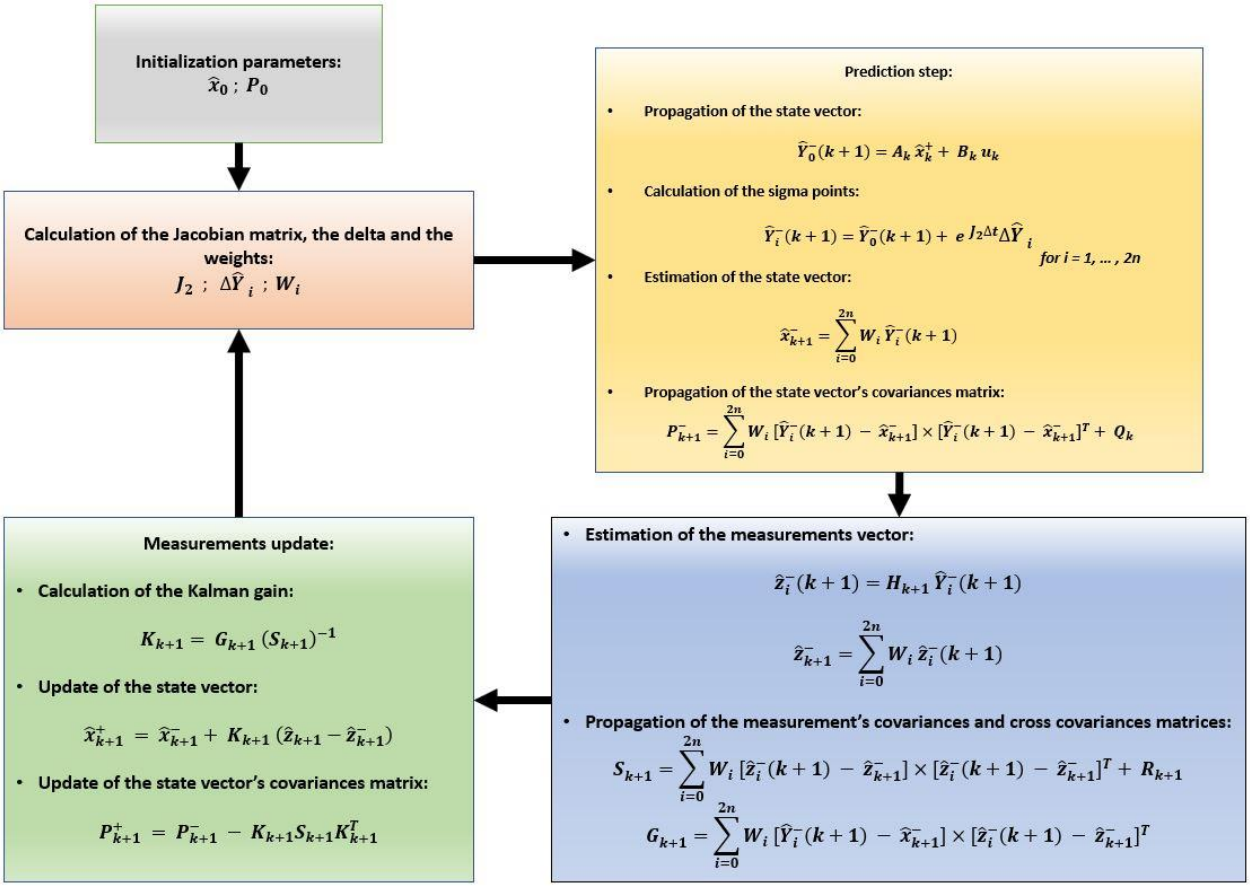

Figure 4.8: Outline of the Extrapolated Single Propagation Unscented Kalman Filter.

In the algorithm of this filter there are not many changes with respect to the algorithm of the SPUKF; indeed, the only difference is the calculation of the ***2n*** Jacobian matrices and their use for the sigma points calculation, and all the other operations are exactly the same of the previous filter.

The principal purpose of these last two filters is the reduction of the computational time of the classic UKF, but their employment can provide also other benefits such as more stability and filtering capacity of the measurements, especially when these measurements are not very accurate.

Chapter 5

Process Model and Filter's set up

The relative translational and rotational dynamics for proximity operations has to be modelled very carefully to design a reliable navigation strategy. The most famous model is the one proposed by Clohessy and Wiltshire, which is an approximated linear model based on the hypothesis of small distance between the spacecraft with respect to their distance from the center of the Earth and of the target spacecraft moving on a circular orbit.

Although this hypothesis could be considered satisfied for the SPARTANS spacecraft simulators, which move in close proximity to each other and belong to a hypothetical circular orbit with the same radius of the Earth, we chose to consider a simpler strategy to describe the two relative dynamics. This decision was made considering that the target spacecraft and the chaser frame are very near to each other, their maximum distance is of about two meters, and mostly because the focus of this thesis is the analysis and comparison of the performances of the developed filters, so it was not necessary to introduce excessive complexity to the model.

The translational motion and the rotational one are independent from each other and so they are analyzed separately, gaining the possibility of employing the linear Kalman filter for the only filtering of the linear translational motion and also the possibility of comparing the developed filters for both a linear and a non-linear system.

Before starting with the filtering process, every filter has to be set up in order to design it for the specific system that it has to analyze. To do so, all the key matrices and parameters must be defined.

For the two motions the used filters are different since the systems to analyze does not have the same nature. Clearly also the set-up of these filters will be different.

For the translational motion the employed filters are: the classic Kalman filter and the linear formulation of the Unscented family filters, the UKF, the SPUKF and the ESPUKF.

For the rotational motion the used filters are: the Extended Kalman filter, the UKF, the SPUKF and the ESPUKF, on their standard non-linear formulation.

5.1 Translational model

For the relative translational motion, we decided to use a linear model based on the standard equation of motion. So, the evolution of the translational kinematic and dynamics is described by:

$$\begin{cases} \dot{x} = v_x + a_x dt \\ \dot{y} = v_y + a_y dt \\ \dot{z} = v_z + a_z dt \end{cases}$$

$$\begin{cases} \dot{v_x} = a_x \\ \dot{v_y} = a_y \\ \dot{v_z} = a_z \end{cases}$$

These equations define the process function $\boldsymbol{f}$ that connects the state vector to its derivative: $\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\hat{\boldsymbol{x}})$.

The propagation chosen for the prediction step of the filter is linear, for both the translational and rotational motion, so the propagation equations are:

$$\begin{cases} x(t+dt) = x(t) + v_x(t)dt + \dfrac{a_x(t)dt^2}{2} \\ y(t+dt) = y(t) + v_y(t)dt + \dfrac{a_y(t)dt^2}{2} \\ z(t+dt) = z(t) + v_z(t)dt + \dfrac{a_z(t)dt^2}{2} \end{cases}$$

$$\begin{cases} v_x(t+dt) = v_x(t) + a_x(t)dt \\ v_y(t+dt) = v_y(t) + a_y(t)dt \\ v_z(t+dt) = v_z(t) + a_z(t)dt \end{cases}$$

These last equations are the expression of the total propagation function $\boldsymbol{F}$ that use the state and the control vectors to propagate the state vector at the next time instant: $\hat{\boldsymbol{x}}^-(\boldsymbol{t}+\boldsymbol{dt}) = \boldsymbol{F}\left(\hat{\boldsymbol{x}}^+(\boldsymbol{t}), \boldsymbol{u}(\boldsymbol{t})\right)$.

For the prediction strategy we have evaluated the possibility to use a different, more structured, propagation from the linear one and it was the *Runge-Kutta 4* propagation. Comparing these two methods there were not valuable differences in the precision of the results, while the computational time of the second method, as it is more complex, was higher. So, in the end, we decided to carry on the analysis employing only the linear propagation.

5.2 Translational set up

The process and the control matrices, ***A*** and ***B***, depend on the model and equations chosen for describing the evolution of the system.

Converting the equations for the translational motion propagation to their matrix form, they become:

$$\begin{Bmatrix} x(t+dt) \\ y(t+dt) \\ z(t+dt) \\ v_x(t+dt) \\ v_y(t+dt) \\ v_z(t+dt) \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ & & 1 & 0 & 0 & dt \\ \vdots & & \ddots & 1 & 0 & 0 \\ & & & & 1 & 0 \\ 0 & & \cdots & & 0 & 1 \end{bmatrix} \begin{Bmatrix} x(t) \\ y(t) \\ z(t) \\ v_x(t) \\ v_y(t) \\ v_z(t) \end{Bmatrix} + \begin{bmatrix} \frac{dt^2}{2} & 0 & 0 \\ 0 & \frac{dt^2}{2} & 0 \\ 0 & 0 & \frac{dt^2}{2} \\ dt & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & dt \end{bmatrix} \begin{Bmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{Bmatrix}$$

This is the prediction equation for the linear translational filters that before was in the implicit form:

$$\hat{x}^-(t+dt) = A(t)\,\hat{x}^+(t) + B(t)\,u(t)$$

So, it is possible to see that:

- The state vector for the translational motion is formed by the relative position and translational velocity between the target and the chaser, while the control vector is formed by the external acceleration acting on the target:

$$\hat{x}^+(t) = \begin{Bmatrix} x(t) \\ y(t) \\ z(t) \\ v_x(t) \\ v_y(t) \\ v_z(t) \end{Bmatrix} \quad ; \quad u(t) = \begin{Bmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{Bmatrix}$$

- The process and the control matrices are very simple:

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{dt} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{dt} & \mathbf{0} \\ & & \mathbf{1} & \mathbf{0} & \mathbf{0} & \boldsymbol{dt} \\ \vdots & & \ddots & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ & & & & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & & \cdots & & \mathbf{0} & \mathbf{1} \end{bmatrix} \quad ; \quad \boldsymbol{B} = \begin{bmatrix} \frac{\boldsymbol{dt^2}}{\mathbf{2}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\boldsymbol{dt^2}}{\mathbf{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\boldsymbol{dt^2}}{\mathbf{2}} \\ \boldsymbol{dt} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{dt} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{dt} \end{bmatrix}$$

It is easy to see that if all the prediction steps follow the same time increment, so the $\boldsymbol{dt}$ is constant, these two matrices are also constant and do not need to be always recalculated. When dealing with measurements coming from the monocular vision camera the time instant will not be constant because it is continuously adapted to the time of the upcoming images that is variable. This introduce a non-linearity and implies the recalculation of these two matrices at every step.

The measurements that are provided to the filters concern only the relative position between the spacecraft, so the measurements estimation equation, $\hat{\boldsymbol{z}}_{\boldsymbol{k+1}}^{-} = \boldsymbol{H}_{\boldsymbol{k+1}}\, \hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-}$, results:

$$\hat{\boldsymbol{z}}^{-}(\boldsymbol{t}+\boldsymbol{dt}) = \begin{Bmatrix} \boldsymbol{x}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{y}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{z}(\boldsymbol{t}+\boldsymbol{dt}) \end{Bmatrix}^{-} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{y}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{z}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{v_x}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{v_y}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{v_z}(\boldsymbol{t}+\boldsymbol{dt}) \end{Bmatrix}^{-}$$

Hence, the measurements matrix $\boldsymbol{H}$ is simply an identity matrix that collects only the first three components of the state vector, the ones that express the relative position.

The last information needed from the classic Kalman filter are the error variances matrices that are defined using the values of the variances calculated by means of the Montecarlo analysis presented before. In the end these matrices assume these expressions:

$$\boldsymbol{P_0} = \boldsymbol{C_P} \begin{bmatrix} \boldsymbol{\sigma_x} & \mathbf{0} & & \cdots & & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\sigma_y} & & & & \\ & & \boldsymbol{\sigma_z} & \ddots & & \vdots \\ \vdots & & \ddots & \boldsymbol{\sigma_{v_x}} & & \\ & & & & \boldsymbol{\sigma_{v_x}} & \mathbf{0} \\ \mathbf{0} & \cdots & & & \mathbf{0} & \boldsymbol{\sigma_{v_x}} \end{bmatrix} \quad ; \quad \boldsymbol{Q} = \boldsymbol{C_Q} \begin{bmatrix} \boldsymbol{\sigma_x} & \mathbf{0} & & \cdots & & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\sigma_y} & & & & \\ & & \boldsymbol{\sigma_z} & \ddots & & \vdots \\ \vdots & & \ddots & \boldsymbol{\sigma_{v_x}} & & \\ & & & & \boldsymbol{\sigma_{v_x}} & \mathbf{0} \\ \mathbf{0} & \cdots & & & \mathbf{0} & \boldsymbol{\sigma_{v_x}} \end{bmatrix}$$

$$\boldsymbol{R} = \boldsymbol{C_R} \begin{bmatrix} \boldsymbol{\sigma_x} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\sigma_y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\sigma_z} \end{bmatrix}$$

$\boldsymbol{P_0}$ is only the initial state vector error covariances matrix, because during the filtering process this matrix is updated at every step according to the equations shown before. For simplicity the process noise covariances matrix, $\boldsymbol{Q}$, is chosen congruent to the previous matrix $\boldsymbol{P_0}$, a part for the factor $\boldsymbol{C_Q}$.

$\boldsymbol{C_P}$, $\boldsymbol{C_Q}$ and $\boldsymbol{C_R}$ are correction factor that are defined during the tuning of the filter. Conducting this operation, the values of this parameters that best tuned the filters were found to be:

- $\boldsymbol{C_P = 1;\ C_Q = 14;\ C_R = 0.1}$; when considering the Motion Capture measurements.
- $\boldsymbol{C_P = 1;\ C_Q = 0.01;\ C_R = 15}$; when considering the monocular camera's measurements.

For the initialization phase the filters require the first matrix of the variances of the state vector, the $\boldsymbol{P_0}$ just seen, and also the first state vector, $\boldsymbol{\hat{x}_0}$. To define this vector, its first three elements, the ones concerning the relative position, are provided by the first measurement coming from the cameras, as in a real case, while the last elements, the ones concerning the relative velocity, are provided from the velocity calculated deriving the position measurements, even though this is not possible in a real case.

Even though the filters of the Unscented family are designed for non-linear system they can be used for linear ones, employing a linear simplification with which the propagation matrices do not need to be recalculated at every step. In this condition, these filters, in addition to the just defined information, need some additional parameters and matrices for their correct set-up.

Firstly, the state vector is augmented with its own error variances:

$$\hat{\boldsymbol{x}}^{+}(\boldsymbol{t}) = \left[\boldsymbol{x(t), y(t), z(t), v_x(t), v_y(t), v_z(t), \sigma_x, \sigma_y, \sigma_z, \sigma_{v_x}, \sigma_{v_y}, \sigma_{v_z}}\right]^{\boldsymbol{T}}$$

Along with this, also all the matrices have to be augmented, but most of them need just the addition of a zeros section in order to satisfy the conditions on the number of elements during the matrices' operations. The three that add significant portions are the $\boldsymbol{A}$, the $\boldsymbol{P_0}$ and the $\boldsymbol{Q}$ matrix as shown here:

$$\boldsymbol{A} = \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ & & 1 & 0 & 0 & dt \\ \vdots & & \ddots & 1 & 0 & 0 \\ & & & & 1 & 0 \\ 0 & & \cdots & & 0 & 1 \end{bmatrix} & \boldsymbol{O_{6\times6}} \\ \boldsymbol{O_{6\times6}} & \boldsymbol{I_{6\times6}} \end{bmatrix}$$

$$\boldsymbol{P_0} = \begin{bmatrix} \boldsymbol{C_P}\begin{bmatrix} \boldsymbol{\sigma_x} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\sigma_y} & & \\ & & \boldsymbol{\sigma_z} & \vdots \\ \vdots & \ddots & \boldsymbol{\sigma_{v_x}} & \\ & & & \boldsymbol{\sigma_{v_x}} \quad \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{\sigma_{v_x}} \end{bmatrix} & \boldsymbol{O_{6\times6}} \\ \boldsymbol{O_{6\times6}} & \boldsymbol{C_P}\begin{bmatrix} \boldsymbol{\sigma_x} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\sigma_y} & & \\ & & \boldsymbol{\sigma_z} & \vdots \\ \vdots & \ddots & \boldsymbol{\sigma_{v_x}} & \\ & & & \boldsymbol{\sigma_{v_x}} \quad \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{\sigma_{v_x}} \end{bmatrix} \end{bmatrix}$$

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{C_Q}\begin{bmatrix} \boldsymbol{\sigma_x} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\sigma_y} & & \\ & & \boldsymbol{\sigma_z} & \vdots \\ \vdots & \ddots & \boldsymbol{\sigma_{v_x}} & \\ & & & \boldsymbol{\sigma_{v_x}} \quad \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{\sigma_{v_x}} \end{bmatrix} & \boldsymbol{O_{6\times6}} \\ \boldsymbol{O_{6\times6}} & \boldsymbol{C_N}\begin{bmatrix} \boldsymbol{\sigma_x} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\sigma_y} & & \\ & & \boldsymbol{\sigma_z} & \vdots \\ \vdots & \ddots & \boldsymbol{\sigma_{v_x}} & \\ & & & \boldsymbol{\sigma_{v_x}} \quad \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{\sigma_{v_x}} \end{bmatrix} \end{bmatrix}$$

Here $\boldsymbol{C_N}$ is the last correction factor and with the tuning this is set to the value of ***0.7***.

Then, the scaling factor $\boldsymbol{\xi}$ has to be defined for the calculation of the sigma points and their weights. This parameter can vary between ***0*** and ***3 – n***, but its variation does not cause any changes in the output of the filters, for the translational motion. So, for this case, it is set to ***3 – n*** that correspond to ***– 9*** because the last three filters, the UKF, the SPUKF and the ESPUKF, employ the augmented state vector and so its elements' number ***n*** is equal to **12**.

At last, for the two Single Propagation Unscented filters, the Jacobian matrix of the process function $\boldsymbol{f}$ must be set up. The form of this matrix is the same for the two filters; the difference stands in the vector used for its evaluation that are the propagated state vector for the SPUKF and the $\boldsymbol{2n}$ semi-augmented state vector for the ESPUKF. However, the general form of this matrix for the translational motion case is really simple and it does not involve any elements of the state vector, so it is exactly the same for both filters and it is constant for every propagation step. Indeed, the matrix $\boldsymbol{J}$ results:

$$\boldsymbol{J} = \begin{bmatrix} & \boldsymbol{dt} & \boldsymbol{0} & \boldsymbol{0} & \\ & \boldsymbol{0} & \boldsymbol{dt} & \boldsymbol{0} & \\ \boldsymbol{O_{12\times6}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{dt} & \boldsymbol{O_{12\times6}} \\ & & \boldsymbol{O_{9\times3}} & & \end{bmatrix}$$

The fact that this matrix is constant implies that the last two filters turn out to be exactly identical. Their formulation is different in the way that they compute the Jacobian matrix, but since that this matrix does not change for any state vector, or augmented state vector, they end up to the same final definitions and therefore to the same results, as it will be shown later.

5.3 Rotational model

For the relative rotational motion model, we chose to express the relative attitude between the spacecraft using attitude quaternions with a scalar-vector notation, where the scalar component is the first element of the quaternion:

$$\boldsymbol{q} = [\, \boldsymbol{q_0}\,, \boldsymbol{q_1}\,, \boldsymbol{q_2}\,, \boldsymbol{q_3}\,]^{\boldsymbol{T}}$$

Hence, naturally, the relative attitude kinematics is described by the quaternion propagation equation:

$$\dot{\boldsymbol{q}} = \frac{\mathbf{1}}{\mathbf{2}}\,[\mathbf{Q}]\,\boldsymbol{\omega}$$

This is the first portion of the process function $\boldsymbol{f}$. Here the propagation matrix $[\mathbf{Q}]$ is defined by the elements of the quaternion:

$$[\mathbf{Q}] = \begin{bmatrix} -\boldsymbol{q_1} & -\boldsymbol{q_2} & -\boldsymbol{q_3} \\ \boldsymbol{q_0} & -\boldsymbol{q_3} & \boldsymbol{q_2} \\ \boldsymbol{q_3} & \boldsymbol{q_0} & -\boldsymbol{q_1} \\ -\boldsymbol{q_2} & \boldsymbol{q_1} & \boldsymbol{q_0} \end{bmatrix}$$

The parametrization of the relative attitude employing quaternions has been chosen since it grants advantages in terms of computational speed and efficiency avoiding the insurgence of singularity that may happen when using the Euler's angles. Clearly, for a better visual rendering and evaluation the results will be displayed using the Euler's angles.

The propagation of the relative attitude dynamics is described the Euler's equations of dynamic:

$$\boldsymbol{I\dot{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I\omega} = \boldsymbol{\tau_e}$$

Here $\boldsymbol{I}$ is the inertia matrix of the target spacecraft, $\boldsymbol{\omega}$ is its absolute angular velocity expressed in the chaser reference frame and $\boldsymbol{\tau_e}$ represents the resultant of all the external control and disturbance torques acting on the target spacecraft. The angular velocity $\boldsymbol{\omega}$, in this case is both the absolute velocity of the target and the relative velocity of the target with respect to the chaser because there is not an actual chaser spacecraft in motion, but it is simulated by a fixed frame that support the monocular vision camera. This is also the reason why there is not the presence of the torques acting on the chaser.

Projecting the equations along the three axes of the body reference frame of the target spacecraft the Euler's equations of dynamic become:

$$\begin{cases} \dot{\omega}_x = \dfrac{(I_y - I_z)\omega_y\omega_z}{I_x} + a_x \\ \dot{\omega}_y = \dfrac{(I_z - I_x)\omega_x\omega_z}{I_y} + a_y \\ \dot{\omega}_z = \dfrac{(I_x - I_y)\omega_x\omega_y}{I_z} + a_z \end{cases}$$

This is the second portion of the process function $\boldsymbol{f}$. Here the component of the inertia matrix, $\boldsymbol{I_x}$, $\boldsymbol{I_y}$ and $\boldsymbol{I_z}$, are the principal moments of inertia of the target spacecraft that are aligned with its body reference frame. The three last parameters, $\boldsymbol{a_x}$, $\boldsymbol{a_y}$ and $\boldsymbol{a_z}$, represent the external angular acceleration acting on the target spacecraft due to the imposed maneuvers and to the friction and external disturbances.

As already mentioned, during the prediction inside the filtering process, the propagation of the motion is linear and so these equations are expressed in this form:

$$\boldsymbol{q(t + dt) = q(t) + \dot{q}(t)\, dt}$$

$$\boldsymbol{\omega(t + dt) = \omega(t) + \dot{\omega}_E(t)\, dt + a_{ext}(t)\, dt}$$

These equations define the total propagation function $\boldsymbol{F}$. Here the angular acceleration $\boldsymbol{\dot{\omega}_E}$ is the variation of the angular velocity defined by the Euler's equation in absence of external forces. This portion of the total angular acceleration is calculated at every time instant during the filters' prediction steps, while the last portion, the one related to the external acceleration, $\boldsymbol{a_{ext}}$, is provided to the filter.

5.4 Rotational set up

The matrix form of the equations chosen for the rotational motion propagation is:

$$\begin{Bmatrix} q_0(t+dt) \\ q_1(t+dt) \\ q_2(t+dt) \\ q_3(t+dt) \\ \omega_x(t+dt) \\ \omega_y(t+dt) \\ \omega_z(t+dt) \end{Bmatrix} = \begin{bmatrix} I_{4\times4} & A_q \\ O_{3\times4} & A_\omega \end{bmatrix} \begin{Bmatrix} q_0(t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \\ \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{Bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ dt & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & dt \end{bmatrix} \begin{Bmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{Bmatrix}$$

where:

$$A_q = \begin{bmatrix} -q_1(t) & -q_2(t) & -q_3(t) \\ q_0(t) & -q_3(t) & q_2(t) \\ q_3(t) & q_0(t) & -q_1(t) \\ -q_2(t) & q_1(t) & q_0(t) \end{bmatrix} \quad ; \quad A_\omega = \begin{bmatrix} 1 & -\omega_z(t)\frac{I_3}{I_1} & \omega_y(t)\frac{I_2}{I_1} \\ \omega_z(t)\frac{I_3}{I_2} & 1 & -\omega_x(t)\frac{I_1}{I_2} \\ -\omega_y(t)\frac{I_2}{I_3} & \omega_x(t)\frac{I_1}{I_3} & 1 \end{bmatrix}$$

So, for this system, the state and the control vectors are:

$$\hat{x}^+(t) = \begin{Bmatrix} q_0(t) \\ q_1(t) \\ q_2(t) \\ q_3(t) \\ \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{Bmatrix} \quad ; \quad u(t) = \begin{Bmatrix} a_x(t) \\ a_y(t) \\ a_z(t) \end{Bmatrix}$$

while the process and control matrices are:

$$A = \begin{bmatrix} I_{4\times4} & A_q \\ O_{3\times4} & A_\omega \end{bmatrix} \quad ; \quad B = \begin{bmatrix} O_{3\times4} \\ dt & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & dt \end{bmatrix}$$

It is possible to see that, in this case, the process matrix has elements that depend on the state vector's component. This is because this system is non-linear and so this matrix needs to be recalculated at every step, given that the state vector changes at every step. It is at this point that some uncertainty may emerge if the approximated linearization has a wrong discretization frequency as already discussed before. However, the non-linearity in these equations are not very high, in fact they concern only directly the elements of the state vector and there are not any differential derivative or other more complex expressions.

The measurements that are provided to the filters concern only the relative attitude between the spacecraft, expressed with the quaternions, so, in this case, the measurements estimation equation, $\hat{\boldsymbol{z}}_{\boldsymbol{k+1}}^{-} = \boldsymbol{H}_{\boldsymbol{k+1}}\, \hat{\boldsymbol{x}}_{\boldsymbol{k+1}}^{-}$, results:

$$\hat{\boldsymbol{z}}^{-}(\boldsymbol{t}+\boldsymbol{dt}) = \begin{Bmatrix} \boldsymbol{q_0}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{q_1}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{q_2}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{q_3}(\boldsymbol{t}+\boldsymbol{dt}) \end{Bmatrix}^{-} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{q_0}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{q_1}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{q_2}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{q_3}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{\omega_x}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{\omega_y}(\boldsymbol{t}+\boldsymbol{dt}) \\ \boldsymbol{\omega_z}(\boldsymbol{t}+\boldsymbol{dt}) \end{Bmatrix}^{-}$$

Hence, the measurements matrix $\boldsymbol{H}$ is simply an identity matrix that collects only the first four components of the state vector.

The Extended Kalman filter needs as last information the error covariances matrices, defined using the values obtained from the Montecarlo analysis. These matrices are:

$$\boldsymbol{P_0} = \boldsymbol{C_P} \begin{bmatrix} \boldsymbol{\sigma_{q_0}} & \mathbf{0} & \cdots & & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\sigma_{q_1}} & & & \\ & & \boldsymbol{\sigma_{q_2}} & \ddots & \vdots \\ \vdots & & \ddots & \boldsymbol{\sigma_{q_3}} & \\ & & & & \boldsymbol{\sigma_{\omega_x}} \\ & & & & & \boldsymbol{\sigma_{\omega_y}} & \mathbf{0} \\ \mathbf{0} & \cdots & & & \mathbf{0} & \boldsymbol{\sigma_{\omega_z}} \end{bmatrix} \quad ; \quad \boldsymbol{Q} = \boldsymbol{C_Q} \begin{bmatrix} \boldsymbol{\sigma_{q_0}} & \mathbf{0} & \cdots & & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\sigma_{q_1}} & & & \\ & & \boldsymbol{\sigma_{q_2}} & \ddots & \vdots \\ \vdots & & \ddots & \boldsymbol{\sigma_{q_3}} & \\ & & & & \boldsymbol{\sigma_{\omega_x}} \\ & & & & & \boldsymbol{\sigma_{\omega_y}} & \mathbf{0} \\ \mathbf{0} & \cdots & & & \mathbf{0} & \boldsymbol{\sigma_{\omega_z}} \end{bmatrix}$$

$$;$$

$$\boldsymbol{R} = \boldsymbol{C_R} \begin{bmatrix} \boldsymbol{\sigma_{q_0}} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\sigma_{q_1}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\sigma_{q_2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\sigma_{q_3}} \end{bmatrix}$$

$\boldsymbol{P_0}$ is only the initial state vector error covariances matrix, because during the filtering this matrix is updated at every step following the equations shown before. Again, for simplicity the process noise covariances matrix, $\boldsymbol{Q}$, is chosen congruent to the previous matrix $\boldsymbol{P_0}$, a part for the factor $\boldsymbol{C_Q}$.

With the tuning of the filters, the correction factors were defined and, only for the Unscented filters, they are different for the two sets of measures:

- For the MC system's measurements, they are as before: $\boldsymbol{C_P = 1}$; $\boldsymbol{C_Q = 14}$; $\boldsymbol{C_R = 0.1}$; $\boldsymbol{C_N = 0.7}$. The EKF uses these values for both cases.

- For the monocular camera's measurements, they are: $\boldsymbol{C_P = 1}$; $\boldsymbol{C_Q = 0.001}$; $\boldsymbol{C_R = 15}$; $\boldsymbol{C_N = 0.7}$.

For the initialization phase, the filters require the first matrix of the covariances of the state vector, the $\boldsymbol{P_0}$ just seen, and also the first state vector, $\hat{\boldsymbol{x}}_{\boldsymbol{0}}$. To define this vector, its first four elements concerning the relative attitude are provided by the first measurement coming from the cameras, as in a real case, while the last elements concerning the relative angular velocity are provided from the rotational velocity calculated deriving the position measurements, even though this is not possible in a real case.

At last there are the additional information needed from the Unscented family filters for their set-up.

The state vector has to be augmented with the error variances of its own elements:

$$\hat{\boldsymbol{x}}^{+}(\boldsymbol{t}) = \left[\boldsymbol{q_0}(\boldsymbol{t}), \boldsymbol{q_1}(\boldsymbol{t}), \boldsymbol{q_2}(\boldsymbol{t}), \boldsymbol{q_3}(\boldsymbol{t}), \boldsymbol{\omega_x}(\boldsymbol{t}), \boldsymbol{\omega_y}(\boldsymbol{t}), \boldsymbol{\omega_z}(\boldsymbol{t}), \boldsymbol{\sigma_{q_0}}, \boldsymbol{\sigma_{q_1}}, \boldsymbol{\sigma_{q_2}}, \boldsymbol{\sigma_{q_3}}, \boldsymbol{\sigma_{\omega_x}}, \boldsymbol{\sigma_{\omega_y}}, \boldsymbol{\sigma_{\omega_z}}\right]^{\boldsymbol{T}}$$

Along with this, the matrices have to be adjusted, adding some zeros portion for some of them. The three that add a significant portion are the $\boldsymbol{A}$, the $\boldsymbol{P_0}$ and the $\boldsymbol{Q}$ matrices in the same way as seen for the previous setting.

Then, the scaling factor $\boldsymbol{\xi}$ has to be defined for the calculation of the sigma points and their weights. This parameter can vary between ***0*** and ***3 – n***, but its variation does not cause any changes in the output of the first two Unscented filters, the UKF and the SPUKF, so, for these ones, it is set to ***3 – n*** corresponding to ***– 11***, because these filters employ the augmented vector that has fourteen elements, so ***n = 14***. For the last filter, the ESPUKF, the variation of this factor actually affects its output and so its performance. Indeed, if this factor is set to ***3 – n***, this filter behaves similarly to the UKF and the SPUKF and so gives also similar output, but when this factor is set to ***0***, the filter unties from the provided measurements. This capacity, even if seems to be positive, can lead this filter to great errors. This happens because, untying from the measurements, the filter pursues its process only trusting its state vector prediction, without updating it with the provided measurements, and so, if there are some errors or noises in the moments of inertia or in the external accelerations provided, the prediction will drift continuously from the actual values without having the possibility to stop and correct the propagation.

At last, for the two Single Propagation Unscented filters, the Jacobian matrix of the process function $\boldsymbol{f}$ must be defined. The form of this matrix is the same for the two filters; the difference stands in the vectors used for its evaluation, which are the propagated state vector for the SPUKF and the $\boldsymbol{2n}$ semi-augmented state vectors for the ESPUKF. So, the expression the Jacobian matrix is:

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J_\omega} & \boldsymbol{J_q} & \boldsymbol{O_{4\times 7}} \\ \boldsymbol{O_{3\times 4}} & \boldsymbol{J_{\dot{\omega}}} & \boldsymbol{O_{3\times 7}} \\ & \boldsymbol{O_{7\times 14}} & \end{bmatrix}$$

where its components are:

$$\boldsymbol{J_\omega} = \begin{bmatrix} \mathbf{0} & -\boldsymbol{\omega_x(t)} & -\boldsymbol{\omega_y(t)} & -\boldsymbol{\omega_z(t)} \\ \boldsymbol{\omega_x(t)} & \mathbf{0} & \boldsymbol{\omega_z(t)} & -\boldsymbol{\omega_y(t)} \\ \boldsymbol{\omega_y(t)} & -\boldsymbol{\omega_z(t)} & \mathbf{0} & \boldsymbol{\omega_x(t)} \\ \boldsymbol{\omega_z(t)} & \boldsymbol{\omega_y(t)} & -\boldsymbol{\omega_x(t)} & \mathbf{0} \end{bmatrix} \quad ; \quad \boldsymbol{J_q} = \begin{bmatrix} -\boldsymbol{q_1(t)} & -\boldsymbol{q_2(t)} & -\boldsymbol{q_3(t)} \\ \boldsymbol{q_0(t)} & -\boldsymbol{q_3(t)} & \boldsymbol{q_2(t)} \\ \boldsymbol{q_3(t)} & \boldsymbol{q_0(t)} & -\boldsymbol{q_1(t)} \\ -\boldsymbol{q_2(t)} & \boldsymbol{q_1(t)} & \boldsymbol{q_0(t)} \end{bmatrix}$$

$$\boldsymbol{J_{\dot{\omega}}} = \begin{bmatrix} \mathbf{0} & \dfrac{\boldsymbol{I_3 - I_2}}{\boldsymbol{I_1}}\boldsymbol{\omega_z(t)} & \dfrac{\boldsymbol{I_3 - I_2}}{\boldsymbol{I_1}}\boldsymbol{\omega_y(t)} \\ \dfrac{\boldsymbol{I_1 - I_3}}{\boldsymbol{I_2}}\boldsymbol{\omega_z(t)} & \mathbf{0} & \dfrac{\boldsymbol{I_1 - I_3}}{\boldsymbol{I_2}}\boldsymbol{\omega_x(t)} \\ \dfrac{\boldsymbol{I_2 - I_1}}{\boldsymbol{I_3}}\boldsymbol{\omega_y(t)} & \dfrac{\boldsymbol{I_2 - I_1}}{\boldsymbol{I_3}}\boldsymbol{\omega_x(t)} & \mathbf{0} \end{bmatrix}$$

5.5 Code optimization

The filters SPUKF and ESPUKF have been developed with the aim of speeding the basic Unscented Kalman filter, but their effect is highly dependent on the number of elements of the state vector, the complexity of the system and the number of prediction steps between each measurement update. In fact, the reduction in computational time is due only to the simplification of the state propagation method.

With some dedicated analyses [11], it is possible to estimate the reduction in time granted by the use of these two filters and, in addition, it is possible to define the limit of the number of elements of the state vector for which the two filters are actually more efficient then the UKF.

The results of these analysis are shown in Fig. 5.1 and Fig. 5.2.

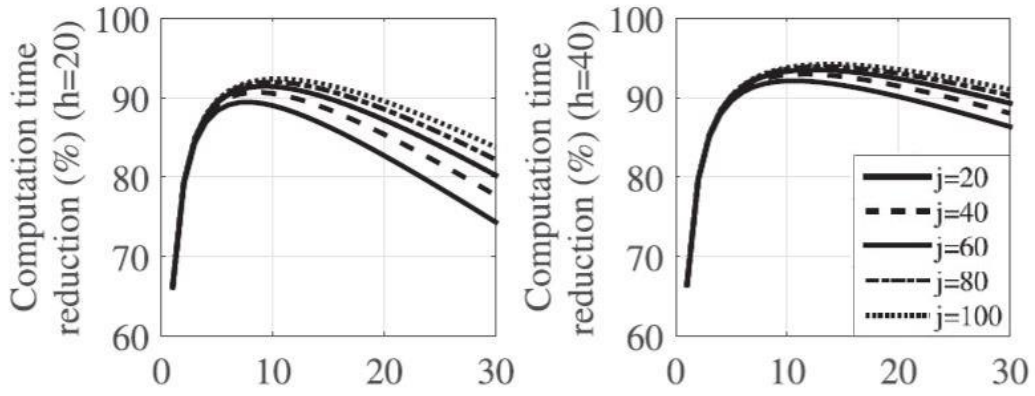

Figure 5.1: Computational efficiency improvements due to the SPUKF implementation as a function of the number of state elements [11].

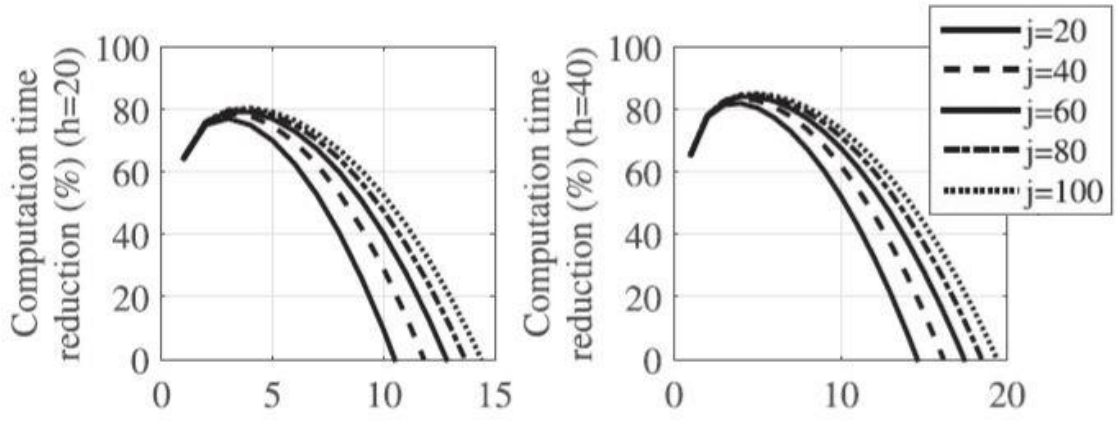

Figure 5.2: Computational efficiency improvements due to the ESPUKF implementation as a function of the number of state elements [11].

The two graphs of Fig. 5.1 refer the SPUKF and the two graphs of Fig. 5.2 refer the ESPUKF. All of them depend on the three mentioned factors: the complexity of the system expressed by the parameter ***j***, which is the number of basic operations required to evaluate the function ***f*** ; the number ***h*** of prediction steps between each measurement update; the number ***n*** of the elements of the state vector.

Considering the linear case of this thesis, i.e. the translational motion, these parameters are listed below.

- ***j = 9***; indeed, for the translational motion the system is linear and very simple.
- ***n = 12***; the state vector elements are the three positions and the three velocities of the center of mass of the target spacecraft with the respect to the chaser and their six relative error variances.
- The number of prediction steps is different depending on the measurements provided to the filters. For the Motion Capture measurements, this number is **100**; for the monocular camera measurements, this number is **10**.

In this case, thanks to the system simplicity, the ESPUKF turns out to be exactly identical to the SPUKF and so only the graphs related to the SPUKF has to be taken into account.

Looking at the trend of the curves in the graphs, the line associated to the considered case will be more pitched toward the right end and its apex will be lower. This is due to the reduced complexity of the system that implies a low value of the parameter ***j***. Considering the number of prediction steps it is easy to see that if this increases, then also the possible time reduction increases. Similarly, if this number decreases also the time reduction decreases.

In the end, it is possible to say that, for the filtering of the linear motion of the spacecraft, the implementation of the SPUKF and ESPUKF filters will supply a reduction on the computational time, in relation to the UKF application, that will be higher in the case with MC measurements.

Considering the non-linear case of this thesis, i.e. the rotational motion, the parameters that influence the time reduction of the single propagation filters are listed in the following.

- ***j = 19***; for the rotational motion the system is more complex than the previous one, even if its complexity is not very high.
- ***n = 14***; the state vector's elements are the four-element quaternion and the three angular velocities of the spacecraft and their seven relative error variances.
- The number of prediction step is still different depending on the measurements provided to the filters. For the Motion Capture measurements, this number is **100**; for the monocular camera measurements, this number is **10**.

In this case the two filters are different, because the Jacobian matrix is dependent from the state vector's elements, and so they follow their related curves.

For the SPUKF the considerations for this case are the same as the ones presented for the previous case, so, in the end, the use of this filter should speed the filtering procedure when compared to the UKF.

For the ESPUKF there are new considerations, because now its effects follow the other curves shown before. The trend of these curves shows that the time reduction provided by this filter is more influenced by the three discussed parameters, mainly the number of state elements. It is easy to see that for ***j = 19*** and ***h = 10*** the associated curve goes below the ***0%*** reduction even for ***n < 10***. So, in the case with monocular camera measurements, where ***h = 10***, since ***n = 14***, there will not be any reduction of time provided by the application of the ESPUKF in place of the UKF, but, instead, there will be an increase in the computational time and so the implementation of this filter will be counter-productive.
Considering the case with MC measurements, where ***h = 100***, it seems that there will be a reduction of time, but its intensity will be much lower than the reduction provided by the SPUKF.

However, it was possible to realize an optimization of the codes of all the Unscented filters. This optimization has been carried out taking advantage of the potentiality of matrix operations of MATLAB. In this way we managed to make simpler and faster the operations of the filters.

For the UKF, the cycles of creation and propagation of the sigma points has been substituted with just some direct matrix operations. This simplification led to an important reduction of the computation time for this filter without any changes in the accuracy, because it represents only a faster way to execute the same operations.

Similarly, also the calculation of the covariances matrices, which is common to all the three Unscented filters, has been made faster.

In the same way we employed some substitution also for the SPUKF and the ESPUKF with matrix operations gaining a reduction on the computational time, but here the intensity of the reduction is lower. This is due to the fact that, in these filters there is the calculation and exponential elevation of the Jacobian matrix at every step and these operations, even when made faster, are still complex and so they appear to be slower compared to the new creation and propagation of the sigma points in the UKF. The operation that require more time to be elaborated is the matrix exponential of the Jacobian matrices. This operation can be pursued applying the MATLAB function ***expm***, but this function is very time-consuming. For the optimization of the two single propagation filters, we created a new function in MATLAB that implements the same operations of the ***expm*** function, but in a much lower time. To define this new function, we exploited the Taylor series expansion of the ***expm*** function and approximated it at its third order, which was the best compromise between precision and time reduction.

In the end, the optimization for the UKF appears to be more effective, such that, for the non-linear case, this filter becomes the faster among the three Unscented ones, while, for the linear case, the other two maintain their highest speed.

However, even if the last two filters do not improve the speed of the UKF in the non-linear case, they can still provide a better accuracy or other characteristics such as more stability and versatility.

Chapter 6

Simulated Maneuvers

The advantage provided by the use of the SPARTANS facility is the possibility to simulate complex maneuvers thanks to the five degree of freedom allowed. For this thesis, we set up four different simulations collecting, from each of them, measurements concerning the relative position and attitude between a moving target spacecraft and a fixed inspector spacecraft.

Among the four simulations, one presents only translational motion, two are characterized only by rotational motion and the last one presents a combined roto-translational motion. For each of these simulations, the collected measurements concern both the translational and the rotational motions; in this way the developed filters can be tested in four different cases for the two types of motions.

The maneuvers were made moving by hand the spacecraft simulator on the surface of the test-table. As far as possible, we provided impulsive forces to move and control the simulator in order to represent a real case of a spacecraft that moves with its thrusters. This was easier to do for the rotational maneuvers, while, for the translational ones, the actions of the forces appear more distributed in time.

The first simulation is associated only to translational maneuvers, but some rotational motions grow thanks to the conservation of the angular momentum, and therefore also in this case it is possible to test the rotational filters. The Figures from 6.1 to 6.4 refer to this simulation.

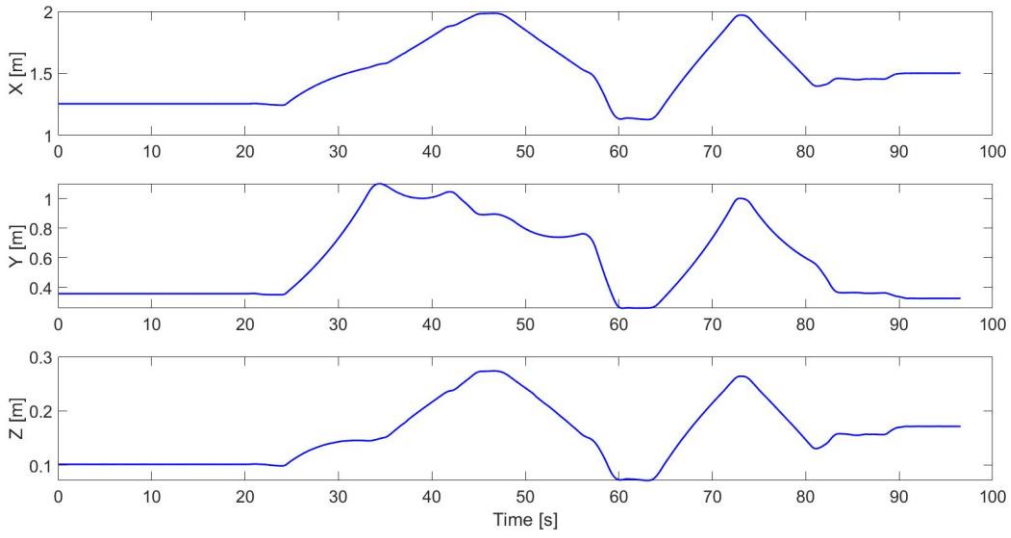

Figure 6.1: Trends of the target's CoM relative position coordinates associated to the first simulation.

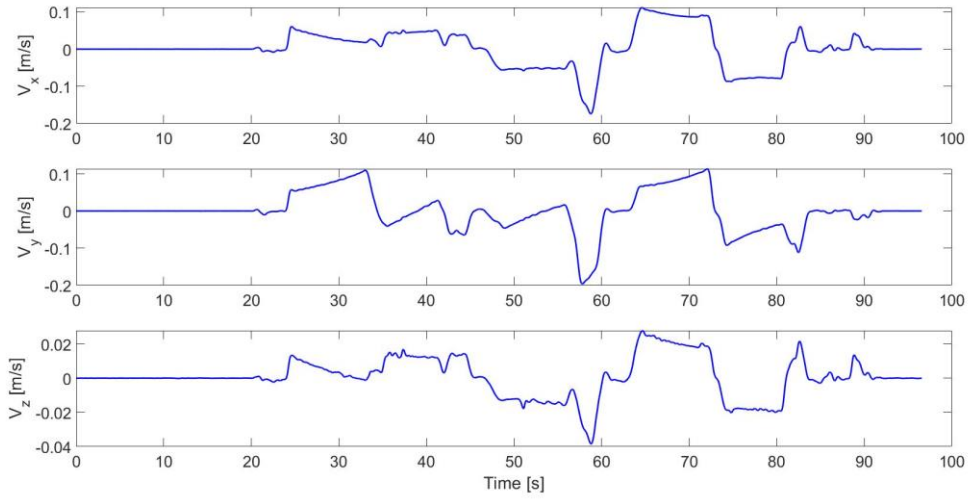

Figure 6.2: Trends of the target's CoM relative translational velocities associated to the first simulation.

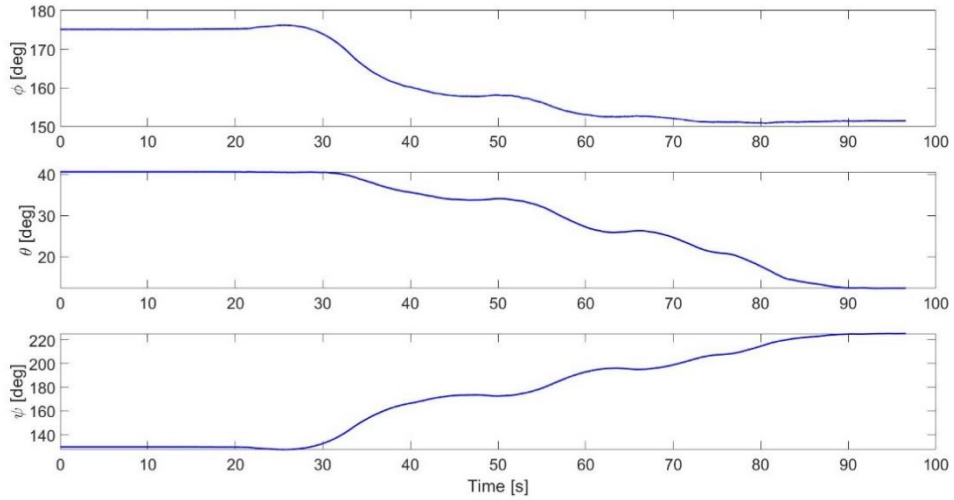

Figure 6.3: Trends of the relative attitude angles associated to the first simulation.

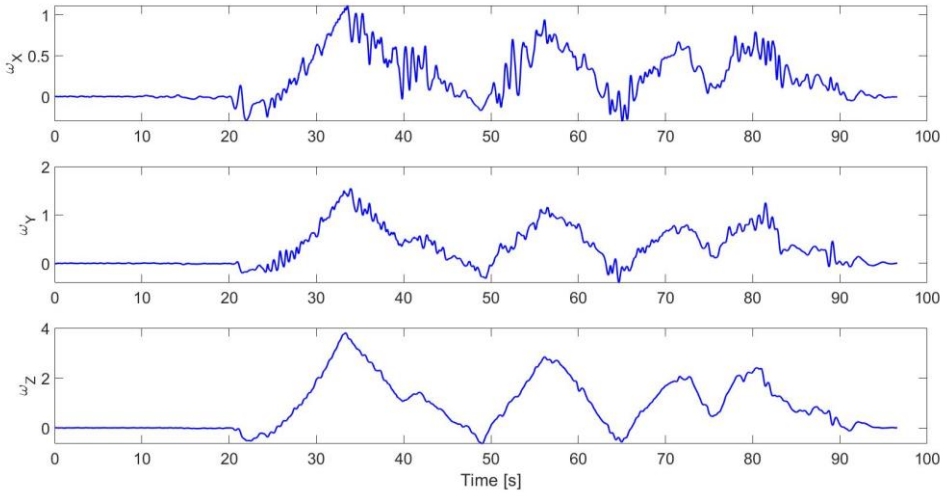

Figure 6.4: Trends of the relative angular velocities associated to the first simulation.

On the other hand, the second and the third simulations concern only the rotational motion. In fact, during these cases the translational module of the simulator was fixed to the test-table and only the attitude module could move rotating around its joint. Even if there is not an actual translational motion, from the translational measurements it is possible to notice small movements of the target spacecraft center of mass. These movements are caused by the uncertainties of the Motion Capture system, indeed their ranges of variation is contained between ± 1 mm. These measurements are still relevant because they can be used for evaluating the performances of the monocular camera with respect to the Motion Capture system.

The trends of the second simulation are shown in the Figures from 6.5 to 6.8.

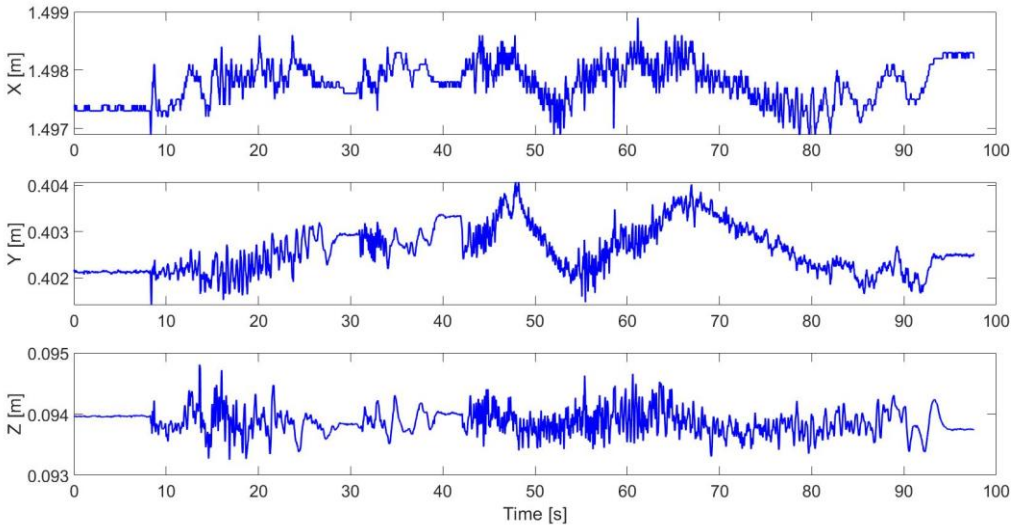

Figure 6.5: Trends of the target's CoM relative position coordinates associated to the second simulation.

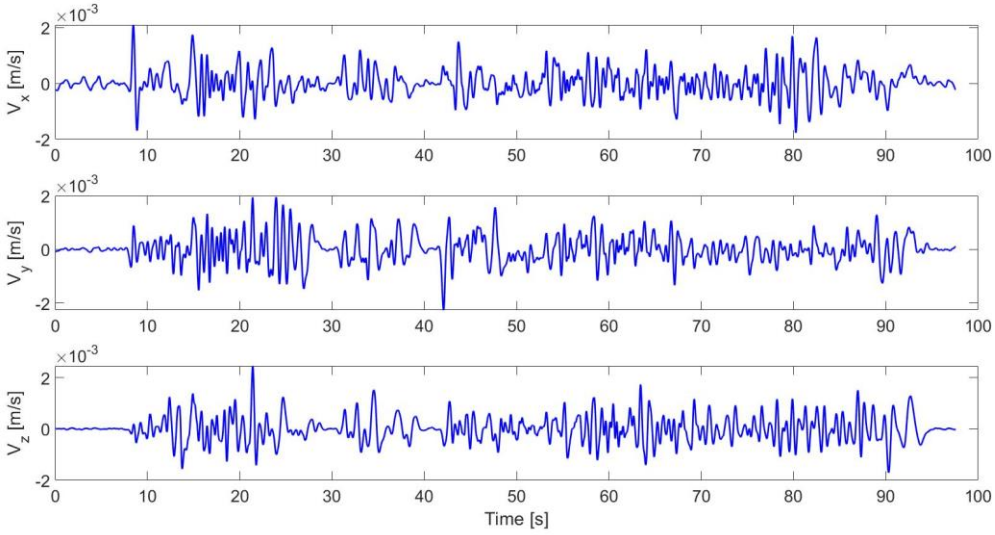

Figure 6.6: Trends of the target's CoM relative translational velocities associated to the second simulation.

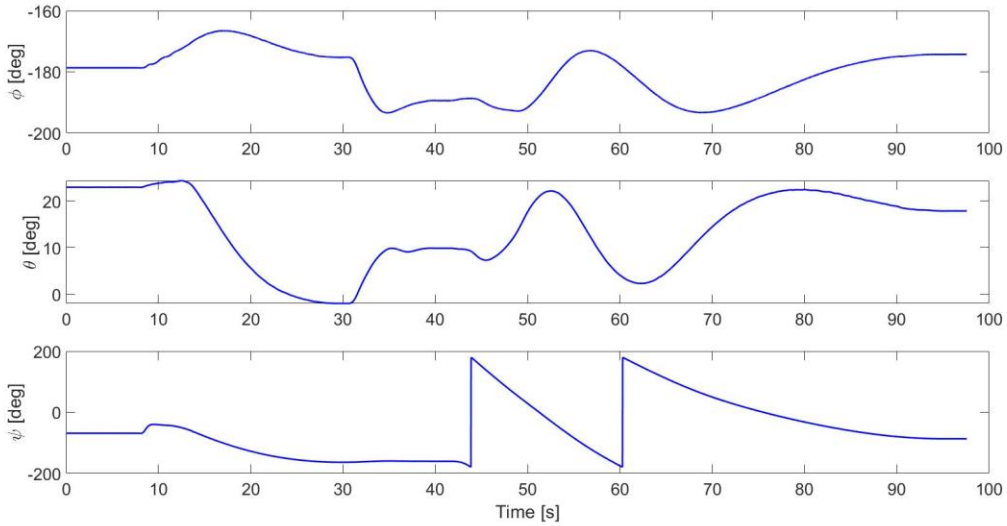

Figure 6.7: Trends of the relative attitude angles associated to the second simulation.

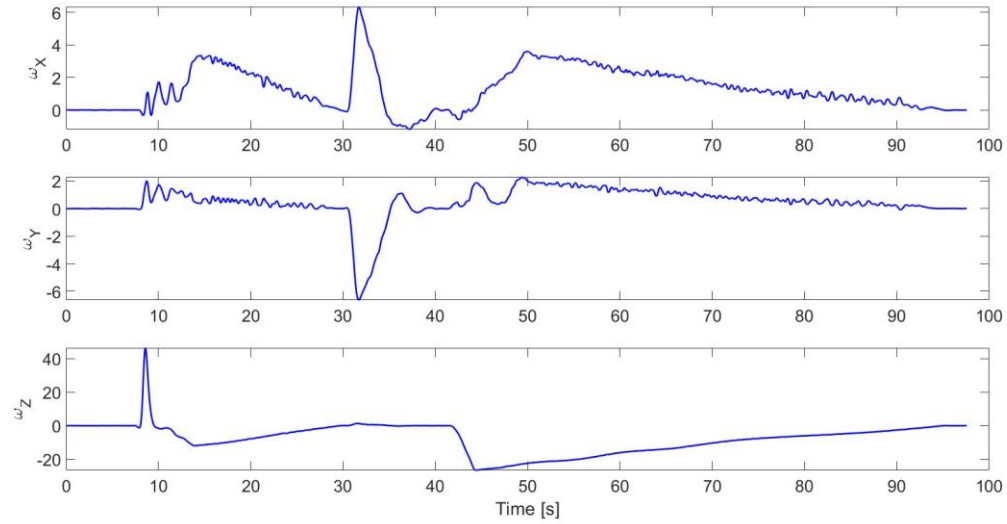

Figure 6.8: Trends of the relative angular velocities associated to the second simulation.

The trends of the third simulation can be seen in the Figures from 6.9 to 6.12.

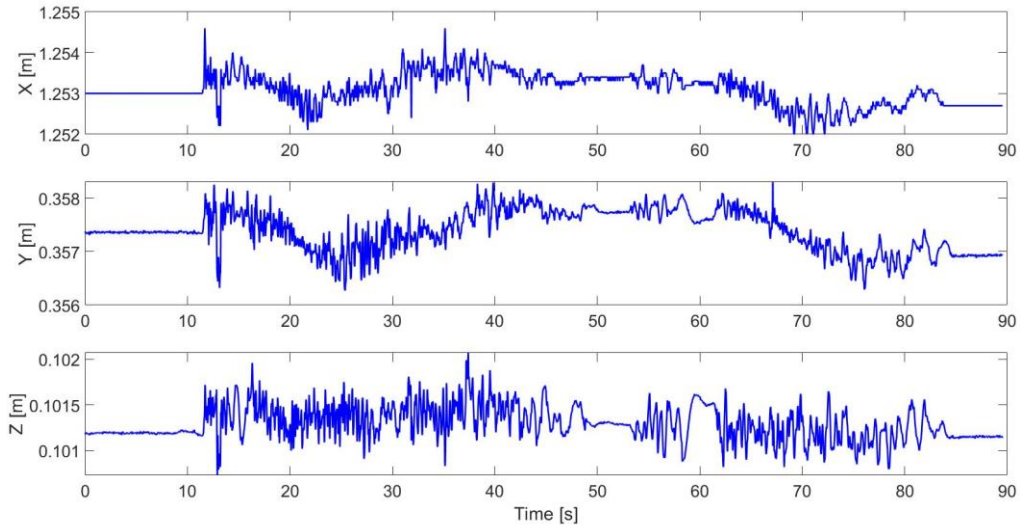

Figure 6.9: Trends of the target's CoM relative position coordinates associated to the third simulation.

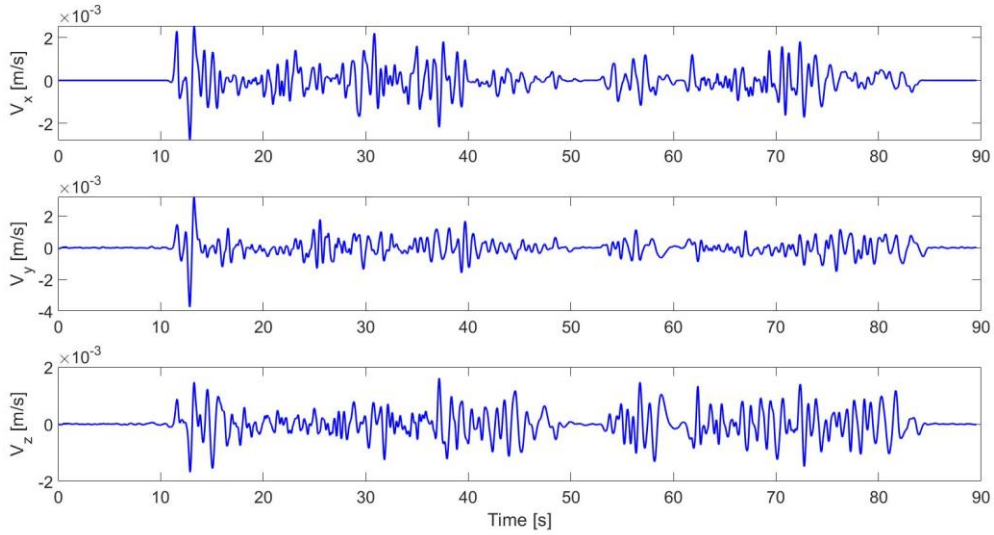

Figure 6.10: Trends of the target's CoM relative translational velocities associated to the third simulation.

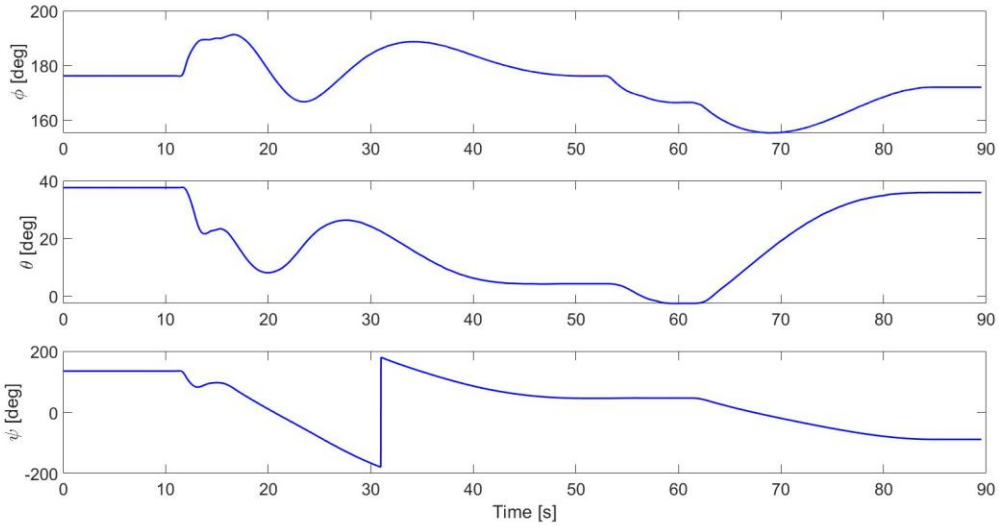

Figure 6.11: Trends of the relative attitude angles associated to the third simulation.

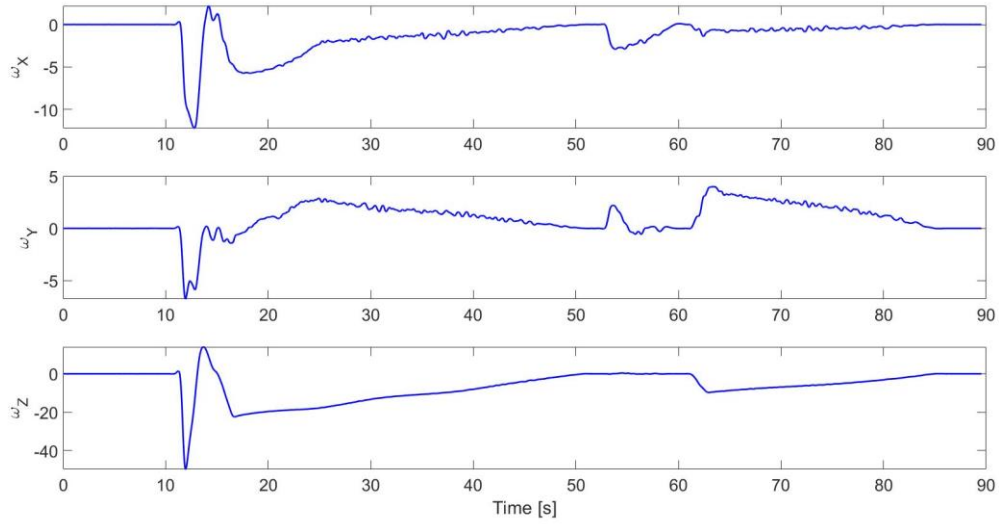

Figure 6.12: Trends of the relative angular velocities associated to the third simulation.

The last acquisition presents a combination of the two motions, so it is the more realistic one and it is suitable for test all the filters in both cases. The Figures from 6.13 to 6.16 concern this simulation.

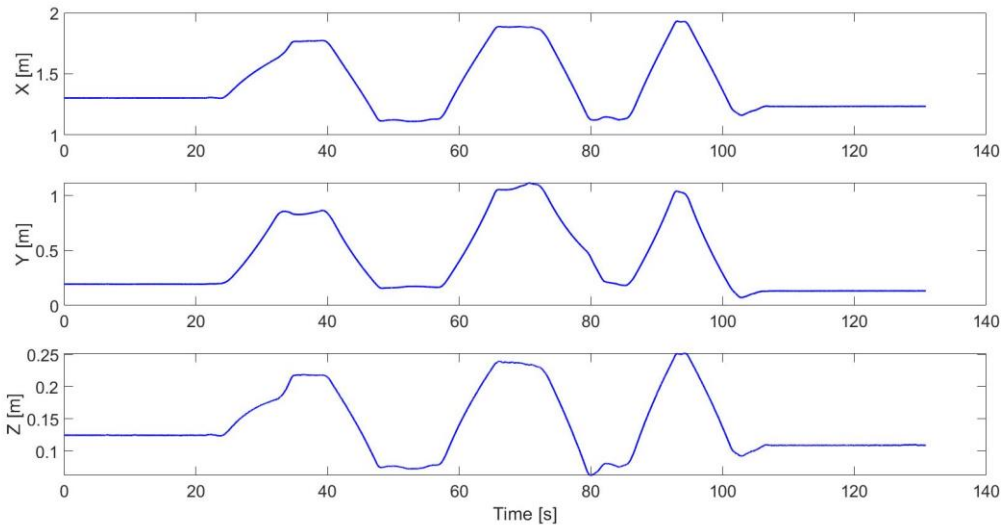

Figure 6.13: Trends of the target's CoM relative position coordinates associated to the fourth simulation.

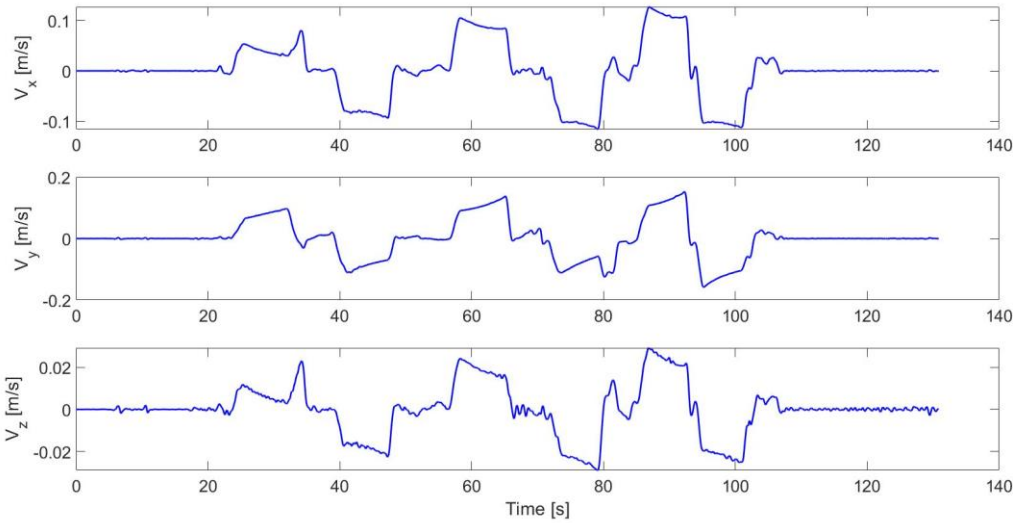

Figure 6.14: Trends of the target's CoM relative translational velocities associated to the fourth simulation.

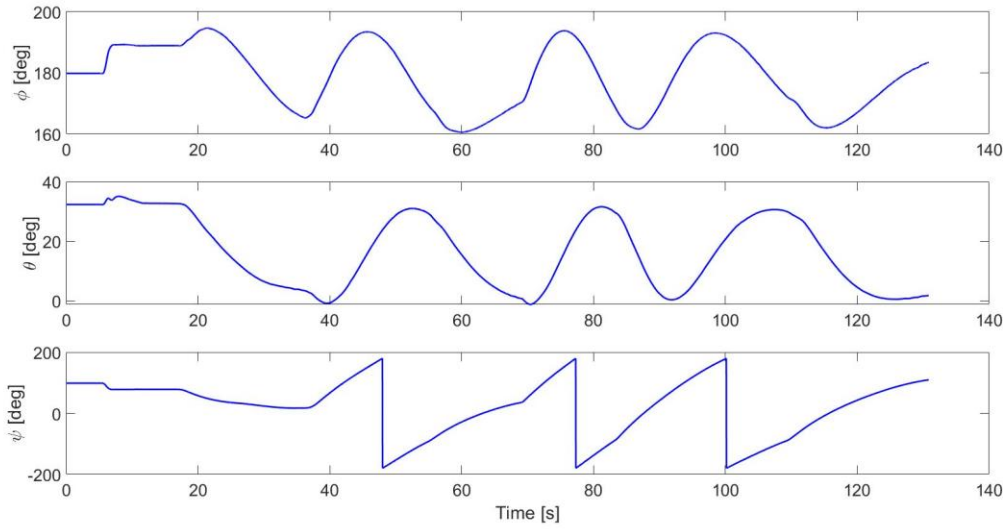

Figure 6.15: Trends of the relative attitude angles associated to the fourth simulation.

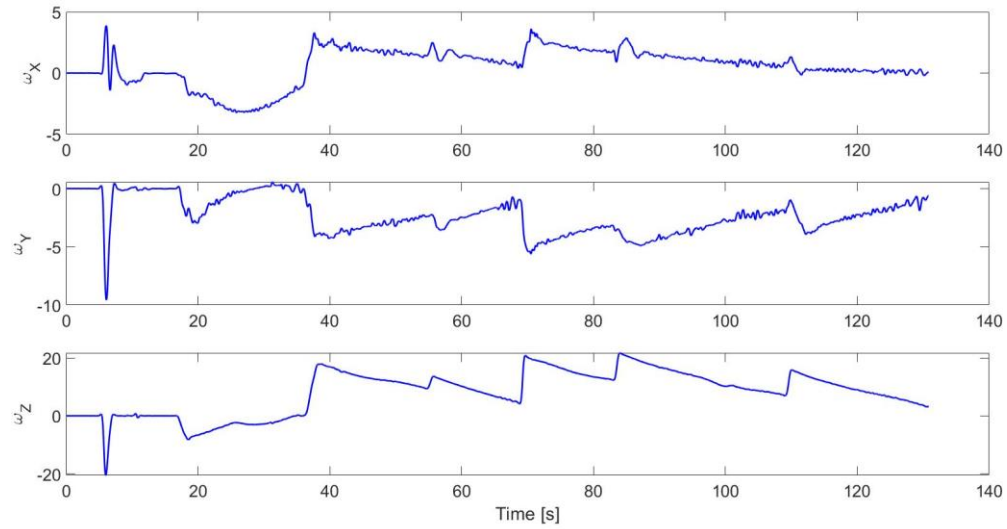

Figure 6.16: Trends of the relative angular velocities associated to the fourth simulation.

The developed filters have been applied to the measurements collected during these simulations to evaluate and compare their performances for the filtering process on a navigation algorithm.

Chapter 7

Filter application on ideal conditions

Firstly, we tested the proposed filters providing them the measurements collected by the MC system. This was made to verify their correct functioning and, meanwhile, see their behavior in an ideal case. These first tests indeed represent an ideal case because all the information provided to the filters are extremely accurate, as described below.

- The measurements provided in this case come from the MC system, but these measurements are also used to define a fiducial reference for the relative dynamics parameters and so this simulates a camera that provide perfect measurements to the spacecraft.
- The external accelerations are exactly the one calculated starting from the measurements and so this simulates a perfect sensor on board of the spacecraft providing this information.
- The moments of inertia are exactly the ones obtained by the SOLIDWORKS model of the SPARTANS simulator and used for the extraction of the Euler's angular accelerations from the global accelerations. In addition to this, the values of these parameters are considered constant in time, even though, in a real case, these could change because of some variations on the target mass or geometry. For these reasons, in the propagation inside the filters there is not the growth of errors due to the uncertainty on the knowledge of the moments of inertia.
- In the end, the first two information, the measurements and the external accelerations, are sent to the filters without any delay with respect to the actual motion, and so all the measurement chain is considered ideal.

One by one these idealizations were removed to, finally, compare the filters for realistic conditions.

To test the filters in this case we provided to them the MC measurements with a lower frequency, of 5 Hz, with respect to their actual one, of 50 Hz. This was done, otherwise the accuracy of the filters' prediction could not be evaluated because the measurements update would be too frequent.

7.1 Translational motion filtering

For the translational motion the employed filters are the linear KF and the three Unscented filters, UKF, SPUKF, ESPUKF, in their linear formulation. As already discussed, in this case, the last two filters are identical and so are their results. For this reason, we will refer only to the SPUKF for their evaluation.

Applying these filters to the four tests carried out in the SPARTANS facility it was possible to collect, as output, the filtered data of the state vector, that are the translational or rotational dynamics parameters. Thanks to the high precision of the provided measurements and to the accuracy of the filters, these filtered data obtained for the four simulations overlap with the reference's data. For this reason, their display will not provide any useful information.
However, there are some difference between the performances of the filters, and these can be appreciated analyzing the mean and the RMS errors between the estimated state vector's elements and the reference's ones throughout the whole acquisitions, and, in addition, evaluating the computational time required from each filter to carry out the analyses. At the end, these results were mediated between the four tests to evaluate more generally the performances of the filters.

The mean and RMS errors on the elements of the state vector for this motion are all shown in the Figures from 7.1 to 7.4 and their related tables.

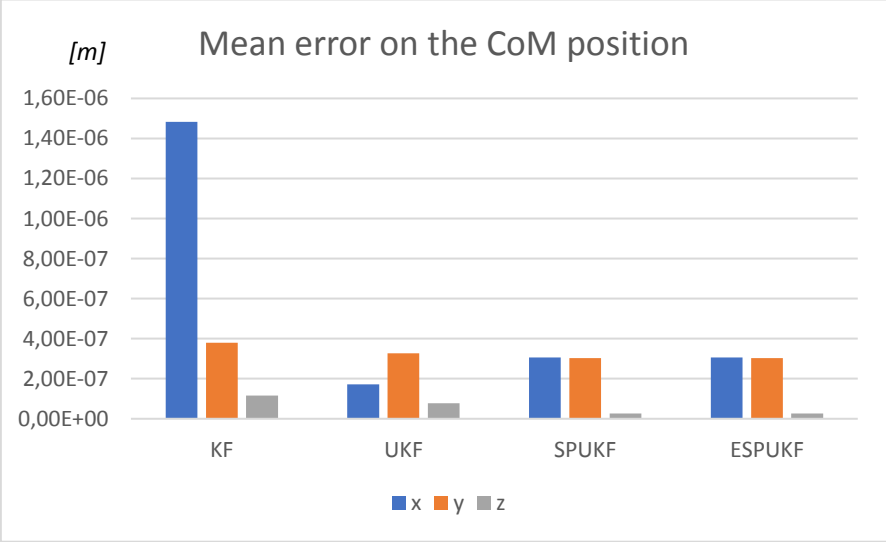

Figure 7.1: Mean errors on the CoM relative position estimated by the filters, mediated over the four tests.

Table 7.1: Mean errors on the CoM relative position estimated by the filters, mediated over the four tests.

Translation Motion MC measurements		**KF**	**UKF**	**SPUKF**	**ESPUKF**
Mean error on the CoM position	**X [m]**	**1,48E-06**	**1,71E-07**	**3,06E-07**	**3,06E-07**
	Y [m]	**3,79E-07**	**3,28E-07**	**3,02E-07**	**3,02E-07**
	Z [m]	**1,16E-07**	**7,79E-08**	**2,63E-08**	**2,63E-08**

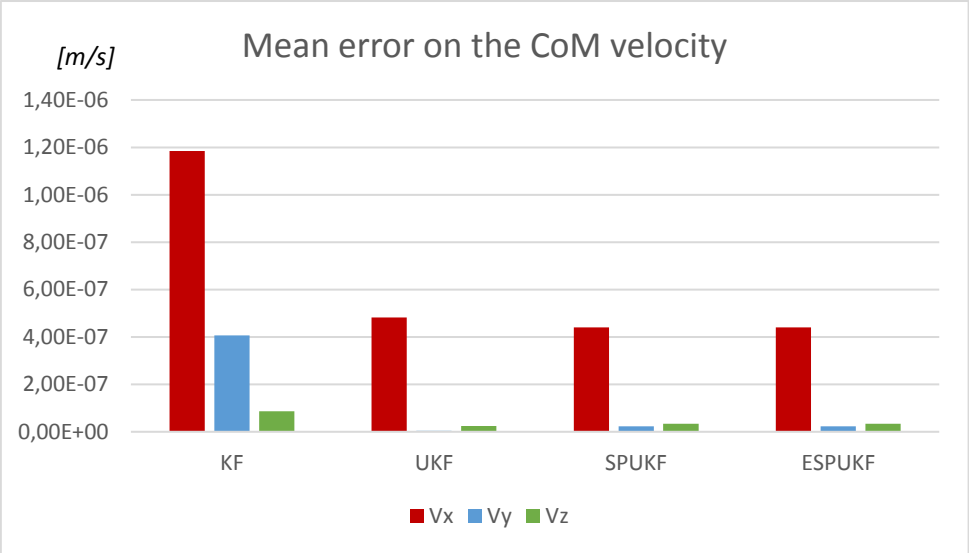

Figure 7.2: Mean errors on the CoM relative velocity estimated by the filters, mediated over the four tests.

Table 7.2: Mean errors on the CoM relative velocity estimated by the filters, mediated over the four tests.

Translation Motion MC measurements		**KF**	**UKF**	**SPUKF**	**ESPUKF**
Mean error on the CoM velocity	**V_X [m/s]**	**1,18E-06**	**4,82E-07**	**4,41E-07**	**4,41E-07**
	V_Y [m/s]	**4,07E-07**	**3,83E-09**	**2,28E-08**	**2,28E-08**
	V_Z [m/s]	**8,59E-08**	**2,49E-08**	**3,42E-08**	**3,42E-08**

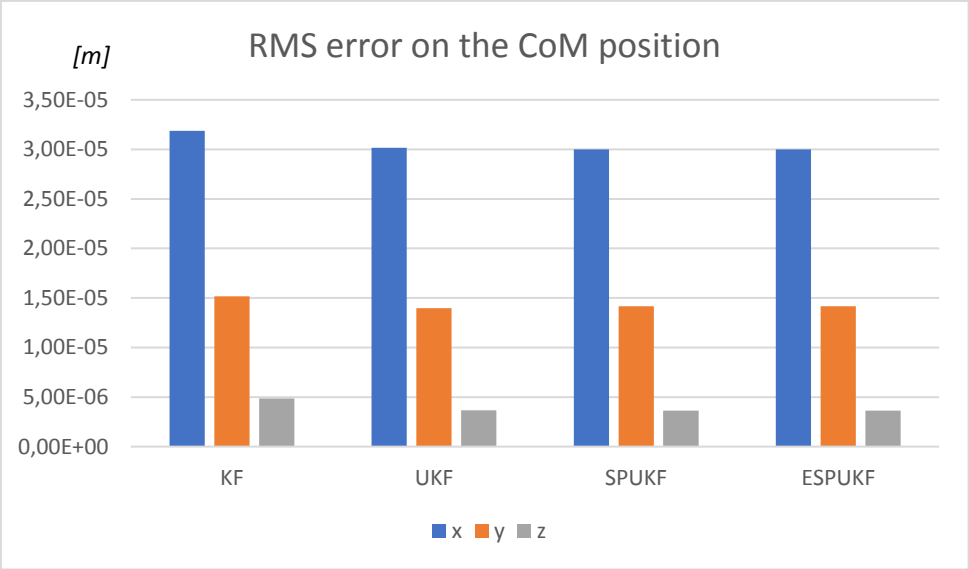

Figure 7.3: RMS errors on the CoM relative position estimated by the filters, mediated over the four tests.

Table 7.3: RMS errors on the CoM relative position estimated by the filters, mediated over the four tests.

Translation Motion MC measurements		**KF**	**UKF**	**SPUKF**	**ESPUKF**
RMS error on the CoM position	**X [m]**	**3,19E-05**	**3,02E-05**	**3,00E-05**	**3,00E-05**
	Y [m]	**1,52E-05**	**1,40E-05**	**1,42E-05**	**1,42E-05**
	Z [m]	**4,84E-06**	**3,67E-06**	**3,65E-06**	**3,65E-06**

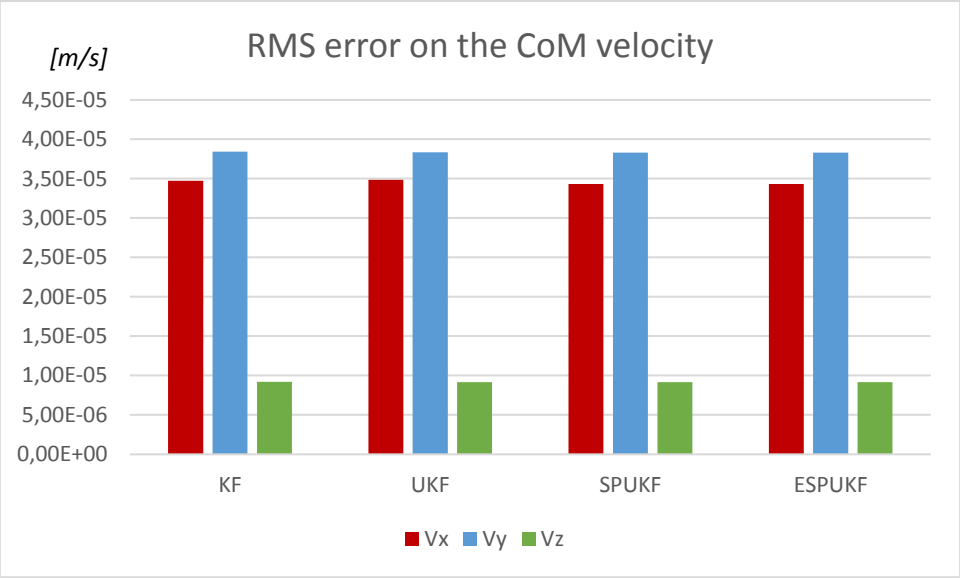

Figure 7.4: RMS errors on the CoM relative velocity estimated by the filters, mediated over the four tests.

Table 7.4: RMS errors on the CoM relative velocity estimated by the filters, mediated over the four tests.

Translation Motion MC measurements		**KF**	**UKF**	**SPUKF**	**ESPUKF**
RMS error on the CoM velocity	**V_X [m/s]**	**3,47E-05**	**3,49E-05**	**3,43E-05**	**3,43E-05**
	V_Y [m/s]	**3,84E-05**	**3,83E-05**	**3,83E-05**	**3,83E-05**
	V_Z [m/s]	**9,20E-06**	**9,16E-06**	**9,16E-06**	**9,16E-06**

From these results it is possible to see that all the filters grant a very high accuracy, but, anyway, the Unscented filters, especially the last two ones, leads to slightly better precision.

To evaluate the computational time of each filter, we launched the analysis of every test employing every filter ten times and then we mediated the values obtained in order to find more reliable results. To measure these CPU times, we used the ***run and time*** feature available in MATLAB.

The mean computation times of each filter for the analysis of the four tests are listed in Table 7.5.

Table 7.5: Mean computational times of each filter for the analysis of the four tests.

CPU time [s]	**KF**	**UKF**	**SPUKF**	**ESPUKF**
Test 1	**23.598**	**28.709**	**27.886**	**27.886**
Test 2	**23.765**	**29.140**	**28.322**	**28.322**
Test 3	**21.739**	**26.259**	**25.637**	**25.637**
Test 4	**32.737**	**39.015**	**38.374**	**38.374**

The tests have different number of measurements that define both the number of measurements update steps and the number of prediction steps, within the filtering process. The number of the prediction steps depends also on the chosen frequency between these two steps; for these analyses we choose to set a hundred propagations between each update. In the end, the number of update and prediction steps for each test are shown in Table 7.6.

Table 7.6: Number of update and prediction steps and their sum for each test.

	Test 1	Test 2	Test 3	Test 4
Update steps	**482**	**487**	**447**	**654**
Prediction steps	**48289**	**48799**	**44779**	**65429**
Total steps	**48771**	**49286**	**45226**	**66083**

Finally, the total CPU time are divided by the sum of the two numbers of steps, in order to evaluate the required time for a single cycle of the filters. These specific computational times for a single filter cycle are listed in Table 7.7 and can be seen in Figure 7.5.

Table 7.7: Computational times for a single cycle of each filter for the analysis of the four tests and their mean.

Single cycle CPU time [ms]	KF	UKF	SPUKF	ESPUKF
Test 1	**0.48385**	**0.58865**	**0.57177**	**0.57177**
Test 2	**0.48219**	**0.59124**	**0.57465**	**0.57465**
Test 3	**0.48068**	**0.58062**	**0.56686**	**0.56686**
Test 4	**0.49539**	**0.59039**	**0.58069**	**0.58069**
Mean	**0.48553**	**0.58773**	**0.57349**	**0.57349**

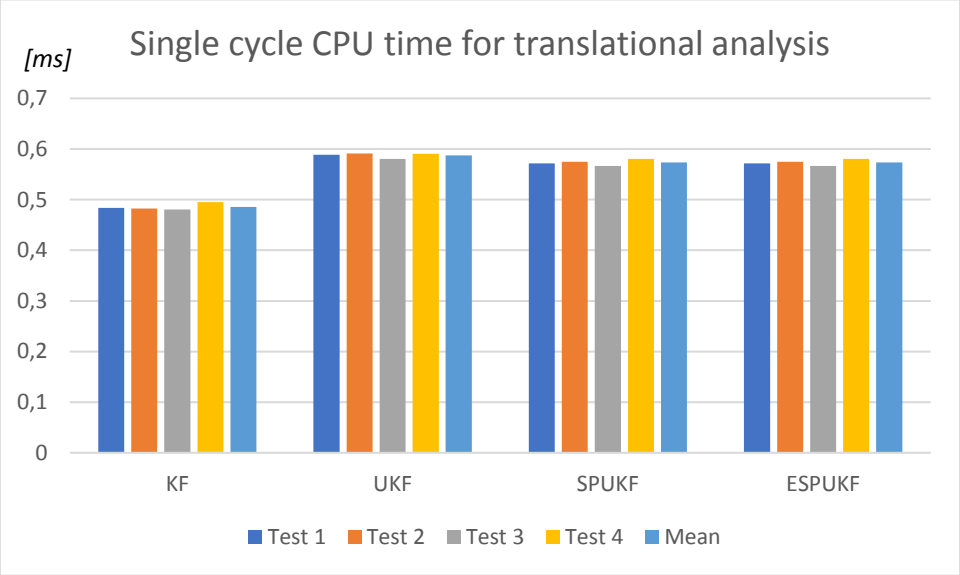

Figure 7.5: Computational times for a single cycle of the filters over all the tests and their mean.

It is possible to see that all the filters appear to be very fast, they just need more or less half a millisecond to fulfil a single cycle. Clearly the linear Kalman filter is the fastest of them, but the difference with the others is not very high. So, for the linear translational motion, the computation time has less importance than the output accuracy when evaluating which filter to employ.
Considering both the CPU time and the mean and RMS errors just shown, the filter that represents the best compromise between these two factors appears to be the SPUKF, because it leads to better accuracy, as compared to the KF, with less computing time than the UKF.

This is only an ideal case, so these considerations are preliminary but useful to verify the correct functioning of all the filters and to have a reference for their foreseen comparison.

7.2 Rotational motion filtering

For the filtering of the rotational motion, the employed filters are the EKF, the UKF, the SPUKF and the ESPUKF, all expressed with their non-linear formulation. In this case the last two filters are not identical and so their results are different.

The provided measurements of the relative attitude are expressed with the quaternions, but the final results, on the mean and RMS error of the output of the filters, are converted in the Euler's angles for an easier understanding of them. The results collected after the application of these filters can be seen in the Figures from 7.6 to 7.9 and in Tables from 7.8 to 7.11.

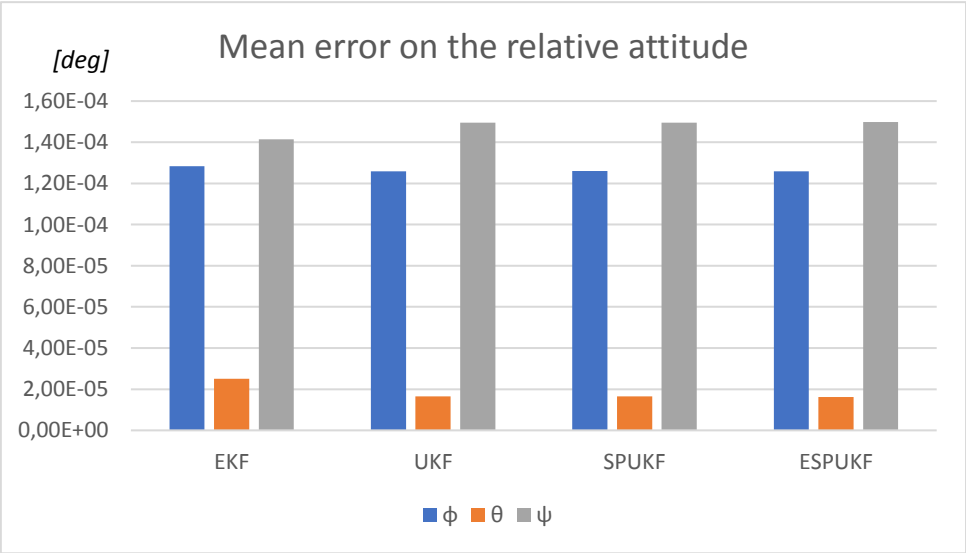

Figure 7.6: Mean errors on the relative attitude estimated by the filters, mediated over the four tests.

Table 7.8: Mean errors on the relative attitude estimated by the filters, mediated over the four tests.

Rotational Motion MC measurements		**EKF**	**UKF**	**SPUKF**	**ESPUKF**
Mean error on relative attitude	**ϕ [deg]**	**1.28E-04**	**1.26E-04**	**1.26E-04**	**1.26E-04**
	θ [deg]	**2.51E-05**	**1.65E-05**	**1.65E-05**	**1.61E-05**
	ψ [deg]	**1.41E-04**	**1.50E-04**	**1.50E-04**	**1.50E-04**

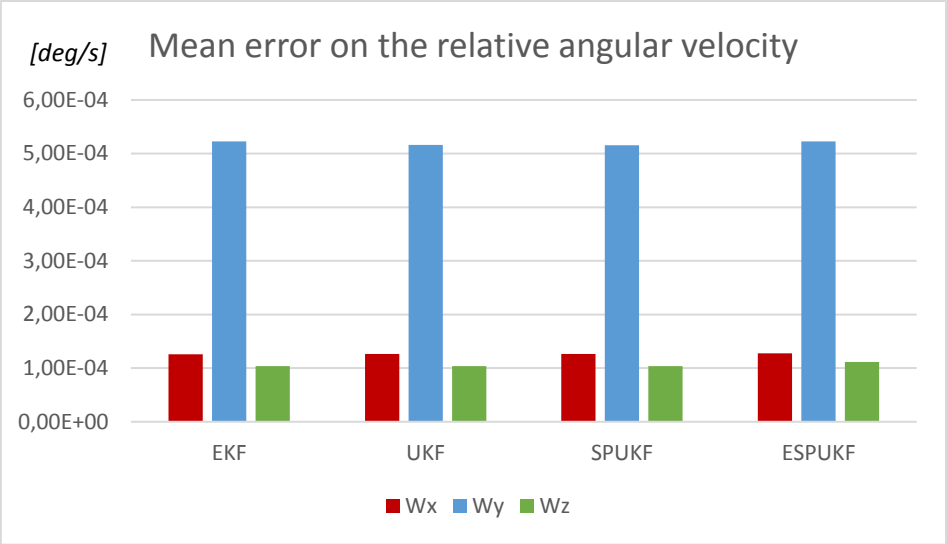

Figure 7.7: Mean errors on the relative angular velocity estimated by the filters, mediated over the four tests.

Table 7.9: Mean errors on the relative angular velocity *estimated by the filters, mediated over the four tests*.

Rotational Motion MC measurements		**EKF**	**UKF**	**SPUKF**	**ESPUKF**
Mean error on relative angular velocity	**ω_X [deg/s]**	**1.26E-04**	**1.26E-04**	**1.27E-04**	**1.28E-04**
	ω_Y [deg/s]	**5.23E-04**	**5.16E-04**	**5.16E-04**	**5.23E-04**
	ω_Z [deg/s]	**1.04E-04**	**1.04E-04**	**1.04E-04**	**1.12E-04**

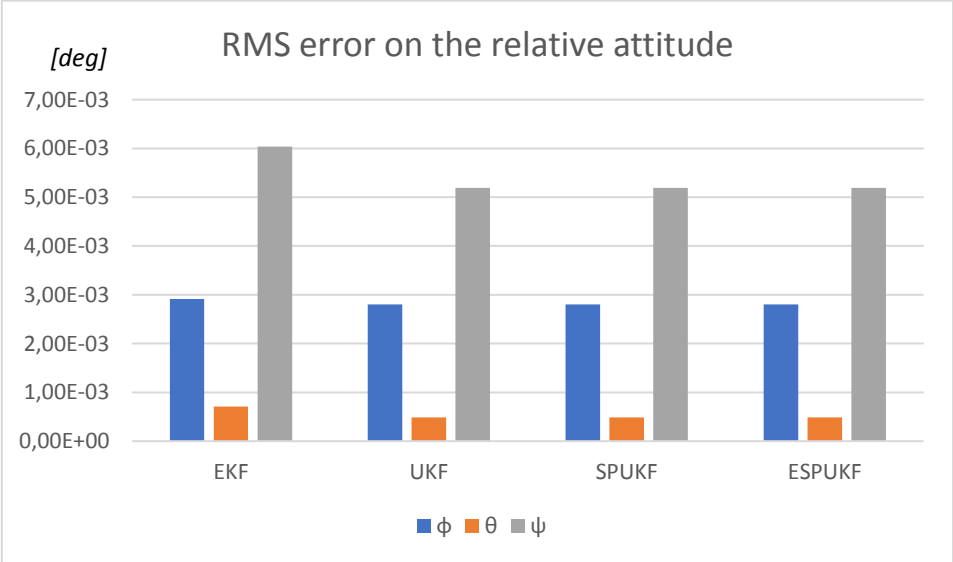

Figure 7.8: RMS errors on the relative attitude estimated by the filters, mediated over the four tests.

Table 7.10: RMS errors on the relative attitude estimated by the filters, mediated over the four tests.

Rotational Motion MC measurements		**EKF**	**UKF**	**SPUKF**	**ESPUKF**
RMS error on relative attitude	**ϕ [deg]**	**2.91E-03**	**2.80E-03**	**2.80E-03**	**2.80E-03**
	θ [deg]	**7.10E-04**	**4.85E-04**	**4.85E-04**	**4.85E-04**
	ψ [deg]	**6.04E-03**	**5.19E-03**	**5.19E-03**	**5.19E-03**

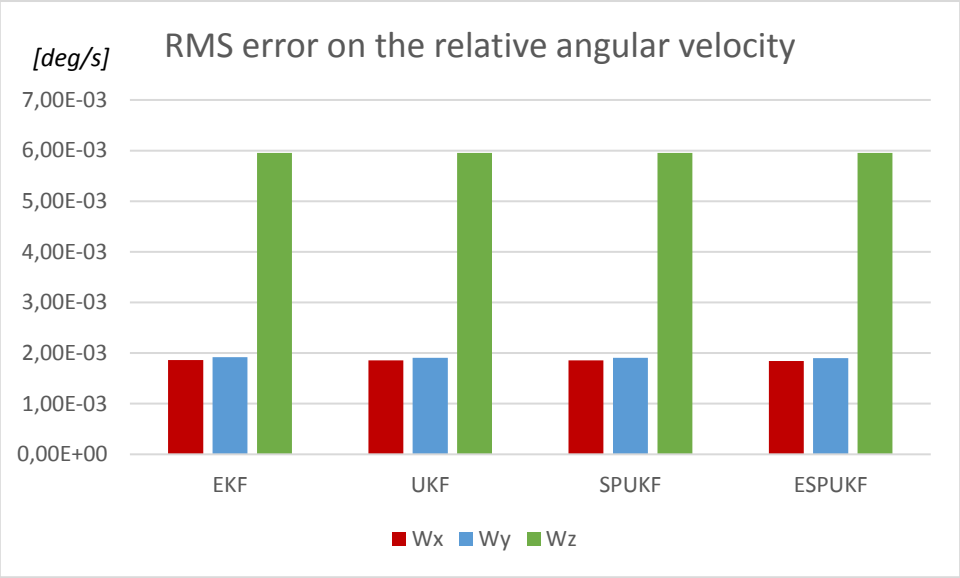

Figure 7.9: RMS errors on the relative angular velocity estimated by the filters, mediated over the four tests.

*Table*7.11: RMS errors on the relative angular velocity *estimated by the filters, mediated over the four tests*.

Rotational Motion MC measurements		**EKF**	**UKF**	**SPUKF**	**ESPUKF**
RMS error on relative angular velocity	**ω_X [deg/s]**	**1.86E-03**	**1.85E-03**	**1.85E-03**	**1.84E-03**
	ω_Y [deg/s]	**1.92E-03**	**1.91E-03**	**1.91E-03**	**1.90E-03**
	ω_Z [deg/s]	**5.96E-03**	**5.95E-03**	**5.95E-03**	**5.96E-03**

Here the differences between the accuracy performances of each filter are lower than in the previous case. So, for the ideal non-linear cases, it is harder to give a preliminary comparison and state the better filter basing only on their accuracy, and therefore the required computational times will gain more importance.

In the same way as done before, we launched ten times the analysis of each test with every filter, registering the computational time employed for each filtering process and then we mediated the obtained values. The mean computational times required by the filters for each test in this case are listed in Table 7.12.

Table 7.12: Mean computational times of each filter for the analysis of the four tests.

CPU time [s]	**EKF**	**UKF**	**SPUKF**	**ESPUKF**
Test 1	**28.483**	**33.781**	**34.587**	**49.683**
Test 2	**29.185**	**34.233**	**35.211**	**50.455**
Test 3	**26.657**	**31.451**	**32.140**	**45.991**
Test 4	**39.545**	**47.177**	**48.176**	**67.180**

The total number of prediction and update steps are the same presented before and shown in Table 7.6.

Finally, the total CPU times are divided by the sum of the two numbers of steps, the prediction and the measurements update steps, in order to evaluate the required time for a single cycle of the filters.

So, the specific computational times for a single cycle of the filters are the ones listed in Table 7.13 and can be seen in Figure 7.10.

Table 7.13: Computational times for a single cycle of each filter for the analysis of the four tests and their mean.

Single cycle CPU time [ms]	EKF	UKF	SPUKF	ESPUKF
Test 1	**0.58402**	**0.69265**	**0.70917**	**1.01870**
Test 2	**0.59216**	**0.69458**	**0.71442**	**1.02372**
Test 3	**0.58942**	**0.69542**	**0.71065**	**1.01692**
Test 4	**0.59841**	**0.71391**	**0.72902**	**1.01660**
Mean	**0.59100**	**0.69914**	**0.71582**	**1.01898**

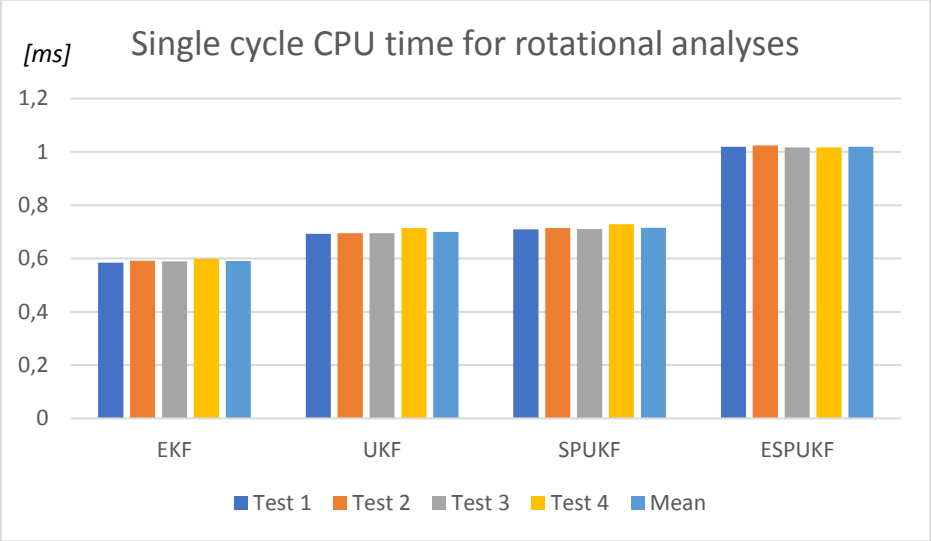

Figure 7.10: Computational times for a single cycle of the filters over all the tests and their mean.

In Fig. 7.10 it is possible to see that the CPU time changes significantly between the filters, the ESPUKF requires almost twice the time of the EKF. So, in this case, the computational time should be carefully evaluated for the choice of the filter to employ.

In the end, for the non-linear rotational motion, there are two best filters depending on which requirement is stricter. If the computational time has more importance than a high accuracy, the filter to be chosen is the standard EKF; if the requirement on the accuracy is more demanding, the best filter is the classic UKF, which leads to a better accuracy than the EKF in the lowest CPU time as compared with the other Unscented filters.

Chapter 8

Filters application on realistic conditions

With the previous analyses, it was possible to compare the filters for ideal situations without the presence of any noise or error. In order to evaluate their performances in more realistic cases, we applied them to the same four tests as before but providing the monocular vision camera's measurements to the filters, thus measurements that a satellite with a vision camera can collect.

To reproduce an even more realistic situation, in the first and the last tests there is the presence of some wrong measures (Fig. 8.1). In this way it was possible also to evaluate which filter react better to this problem (Figures from 8.2 to 8.5).

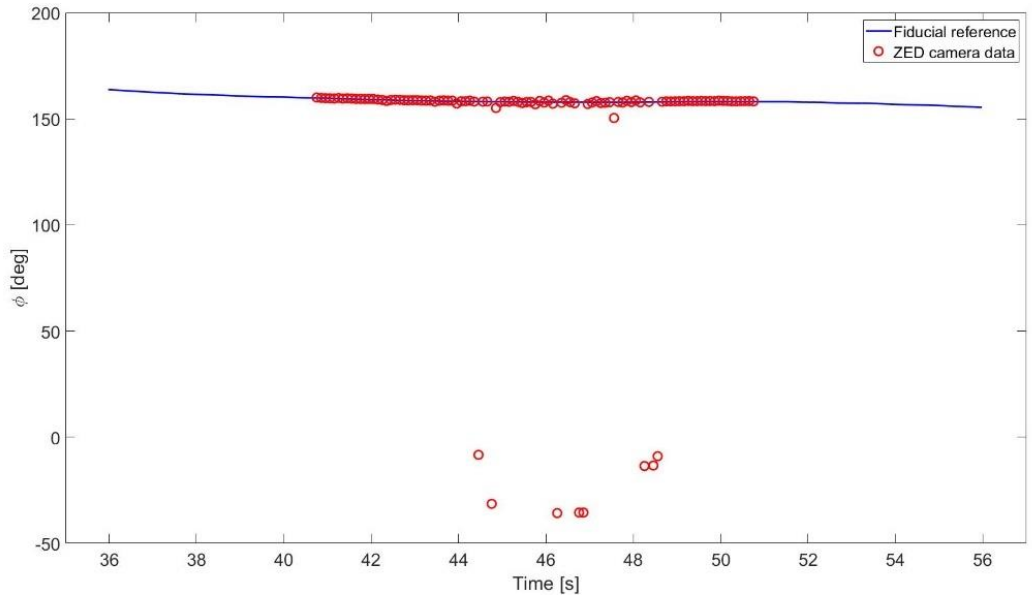

Figure 8.1: Example of the new provided measurements with some wrong data (the red circles), compared to the fiducial reference (the blue line).

The filtered data obtained in this case does not overlap with the reference's data, especially near the wrong measurements, but they are still quite accurate. For this reason, the more interesting comparison is the one displayed in the next figures, from 8.2 to 8.5, were it is possible to see the reaction of the filters to the wrong provided measurements.

For the estimation of the translational motion of the fourth test the worst and the best performances are associated respectively to the linear Kalman filter and the SPUKF.

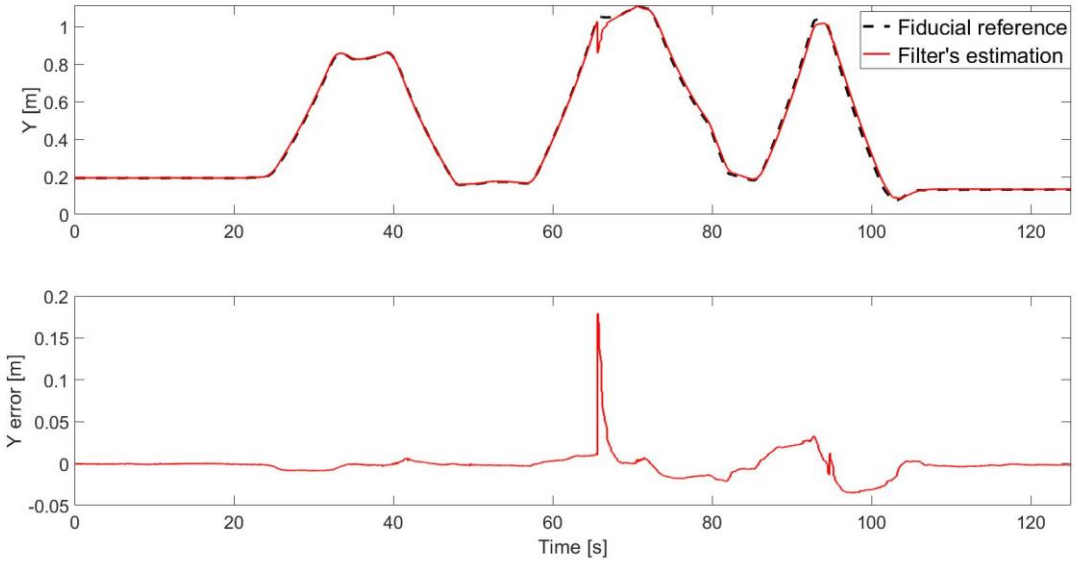

Figure 8.2: Estimation and error of the relative y coordinate of the Target CoM provided by the linear Kalman filter.

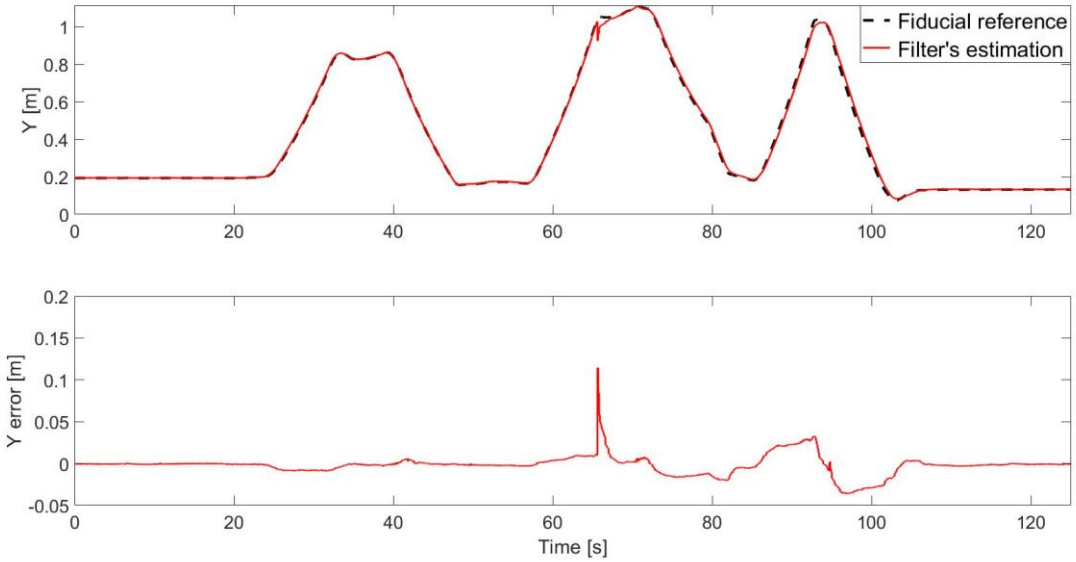

Figure 8.3: Estimation and error of the relative y coordinate of the target CoM provided by the SPUKF.

For the estimation of the rotational motion of the first simulation the worst and the best performances are associated respectively to the Extended Kalman Filter and the ESPUKF.

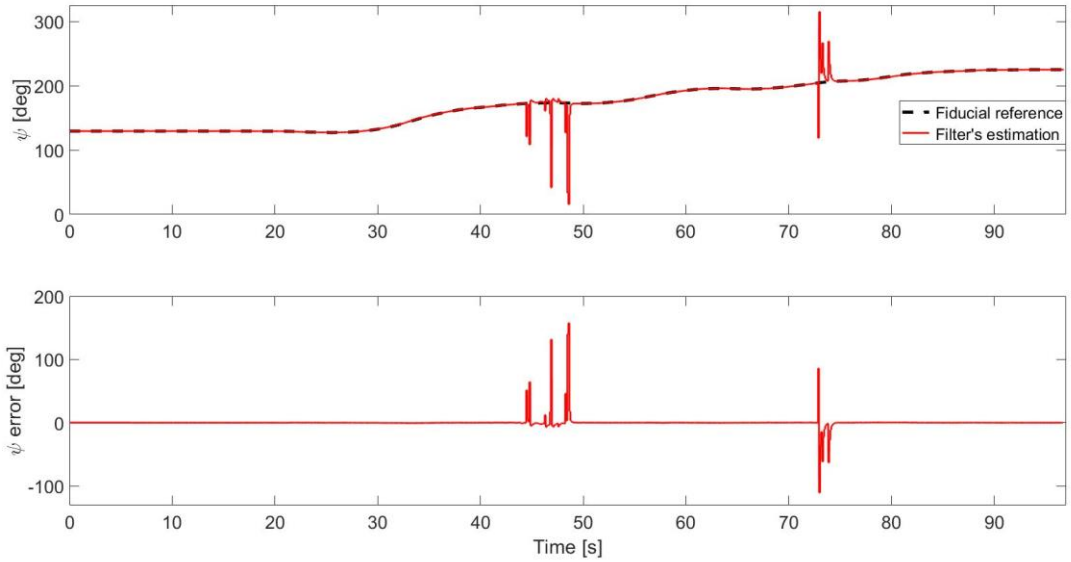

Figure 8.4: Estimation and error of the relative yaw angle provided by the EKF.

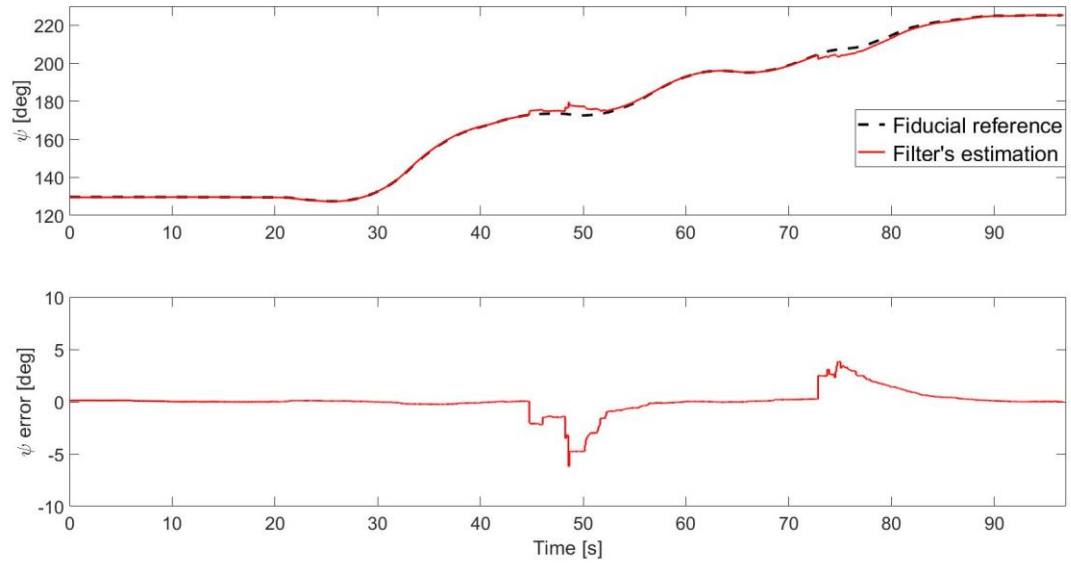

Figure 8.5: Estimation and error of the relative yaw angle provided by the ESPUKF.

Just looking at these figures (8.2 - 8.5) it is possible to see that the Unscented filters can work better even in presence of wrong measurements, especially for the non-linear rotation motion, while the standard filters, the KF and the EKF, appear more affected. However, the actual performances of the filters can be better analyzed comparing their mean and RMS errors as done in the previous chapter.

It is important to remember that, for the translational motion, the SPUKF and the ESPUKF represent the same filter since they do not show any difference, so, we refer only to the SPUKF for their evaluation.

8.1 Translational motion

Applying the linear filters to the data coming from the monocular camera, we collected the results for the mean and RMS errors for the estimated data with respect to their reference shown in the Figures from 8.6 to 8.9 and in the Tables from 8.1 to 8.4.

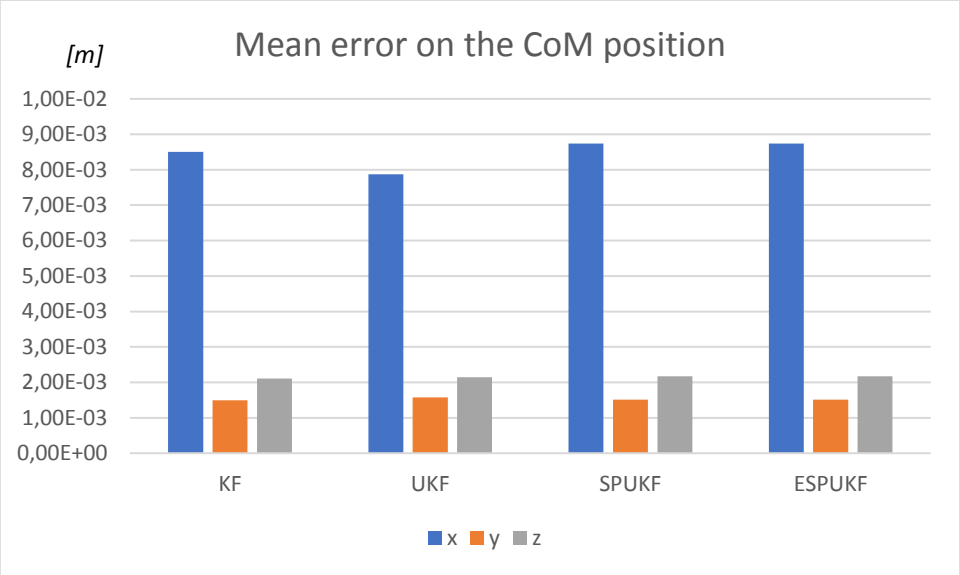

Figure 8.6: Mean errors on the CoM relative position estimated by the filters, mediated over the four tests.

Table 8.1: Mean errors on the CoM relative position estimated by the filters, mediated over the four tests.

Translation Motion Monocular camera measurements		**KF**	**UKF**	**SPUKF**	**ESPUKF**
Mean error on the CoM position	**X [m]**	**8.50E-03**	**7.87E-03**	**8.74E-03**	**8.74E-03**
	Y [m]	**1.49E-03**	**1.58E-03**	**1.51E-03**	**1.51E-03**
	Z [m]	**2.10E-03**	**2.14E-03**	**2.17E-03**	**2.17E-03**

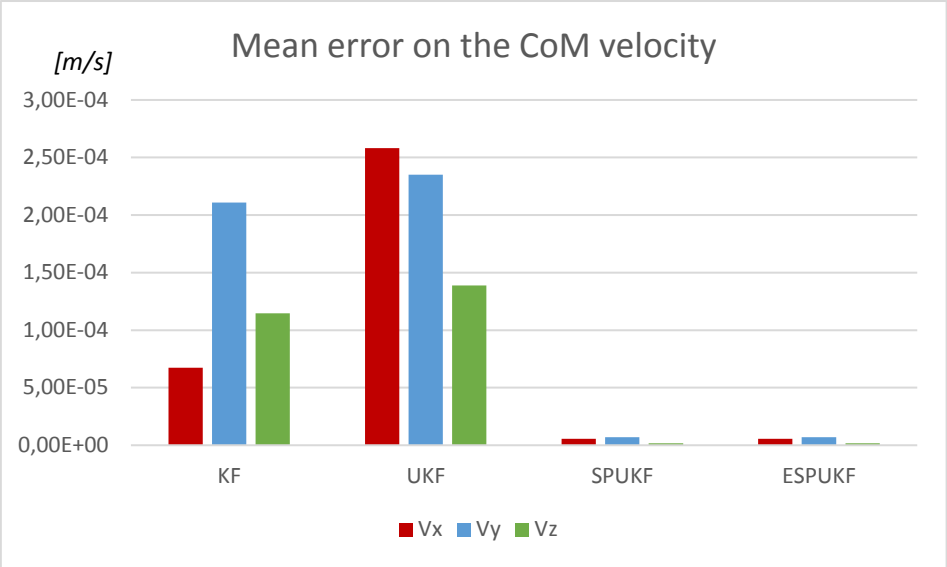

Figure 8.7: Mean errors on the CoM relative velocity estimated by the filters, mediated over the four tests.

Table 8.2: Mean errors on the CoM relative velocity estimated by the filters, mediated over the four tests.

Translation Motion Monocular camera measurements		**KF**	**UKF**	**SPUKF**	**ESPUKF**
Mean error on the CoM velocity	**V_X [m/s]**	**6.74E-05**	**2.58E-04**	**5.48E-06**	**5.48E-06**
	V_Y [m/s]	**2.11E-04**	**2.35E-04**	**7.04E-06**	**7.04E-06**
	V_Z [m/s]	**1.15E-04**	**1.39E-04**	**1.79E-06**	**1.79E-06**

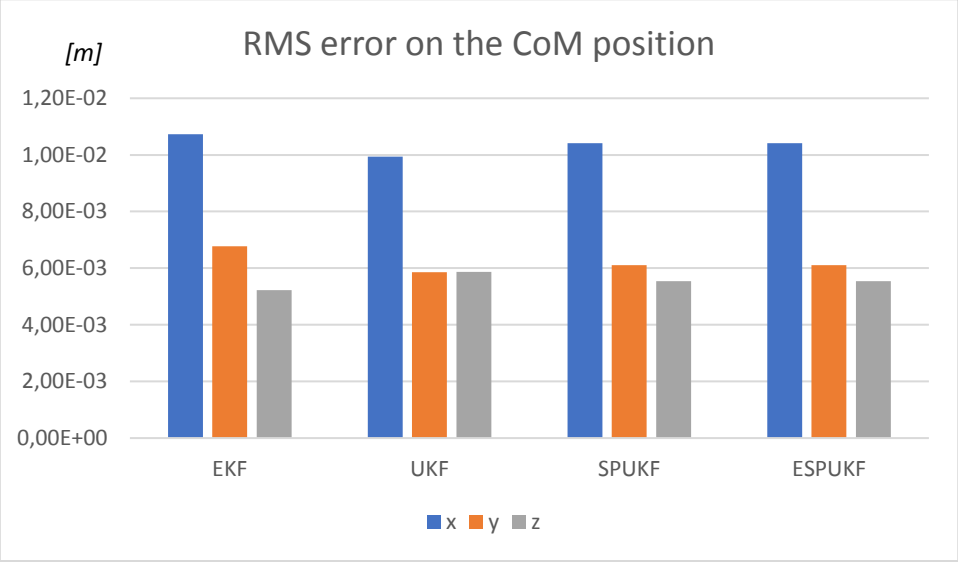

Figure 8.8: RMS errors on the CoM relative position estimated by the filters, mediated over the four tests.

Table 8.3: RMS errors on the CoM relative position estimated by the filters, mediated over the four tests.

Translation Motion Monocular camera measurements		**KF**	**UKF**	**SPUKF**	**ESPUKF**
RMS error on the CoM position	**X [m]**	**1.07E-02**	**9.94E-03**	**1.04E-02**	**1.04E-02**
	Y [m]	**6.77E-03**	**5.85E-03**	**6.11E-03**	**6.11E-03**
	Z [m]	**5.22E-03**	**5.87E-03**	**5.54E-03**	**5.54E-03**

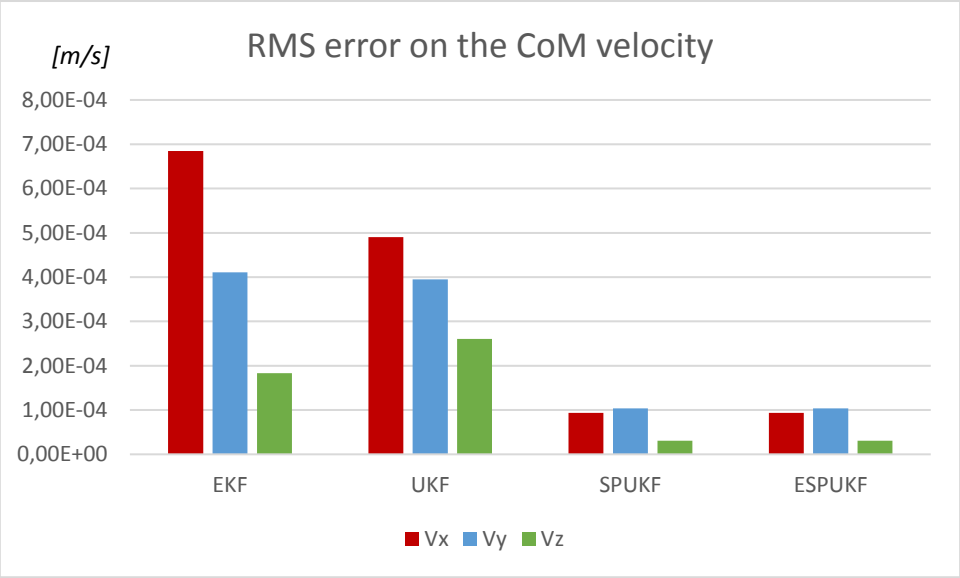

Figure 8.9: RMS errors on the CoM relative velocity estimated by the filters, mediated over the four tests.

Table 8.4: RMS errors on the CoM relative velocity estimated by the filters, mediated over the four tests.

Translation Motion Monocular camera measurements		**KF**	**UKF**	**SPUKF**	**ESPUKF**
RMS error on the CoM velocity	**V_X [m/s]**	**6.85E-04**	**4.90E-04**	**9.35E-05**	**9.35E-05**
	V_Y [m/s]	**4.11E-04**	**3.95E-04**	**1.04E-04**	**1.04E-04**
	V_Z [m/s]	**1.83E-04**	**2.60E-04**	**3.06E-05**	**3.06E-05**

From these results it is clear that all the filters have high accuracy and a good capacity of reacting to the presence of some wrong data. In fact, the mean and the RMS errors stand between a centimeter for the relative position and less than a millimeter per second for the relative velocity, considering a baseline of two meters for the position and of less than twenty centimeters per second for the velocity.

Looking at the estimation of the relative position, all the four filters have almost the same accuracy. On the other hand, evaluating the relative velocity, it is easy to identify the most accurate filter, which is the SPUKF, as noticed before in the ideal conditions.

In this case the computational times are different from the ones of the previous analyses. In fact, launching ten times the analysis of each filter for every test, measuring its required CPU time and then mediating them, the new obtained values are listed in Table 8.5.

Table 8.5: Mean computational times of each filter for the analysis of the four tests.

CPU time [s]	**KF**	**UKF**	**SPUKF**	**ESPUKF**
Test 1	**6.6748**	**8.0008**	**7.8850**	**7.8850**
Test 2	**6.8452**	**8.0066**	**7.9665**	**7.9665**
Test 3	**6.3581**	**7.3794**	**7.2967**	**7.2967**
Test 4	**8.7656**	**10.275**	**10.027**	**10.027**

For these analyses, also the number of prediction and update steps are different. This because the number of the provided measurements is different, and the prediction is pursued ten times between each update instead of a hundred times. So, the number of update and prediction steps for each test are the ones shown in Table 8.6.

Table 8.6: Number of update and prediction steps and their sum for each test.

	Test 1	Test 2	Test 3	Test 4
Update steps	**910**	**976**	**861**	**1231**
Prediction steps	**10091**	**10227**	**9294**	**13053**
Total steps	**11001**	**11203**	**10155**	**14284**

Dividing the total CPU time with the total number of steps it is possible to calculate the computation times for a single filter cycle, which are listed in Table 8.7 and shown in Figure 8.10.

Table 8.7: Computational times for a single cycle of each filter for the analysis of the four tests and their mean.

Single cycle CPU time [ms]	KF	UKF	SPUKF	ESPUKF
Test 1	**0.60675**	**0.72728**	**0.71675**	**0.71675**
Test 2	**0.61102**	**0.71468**	**0.71110**	**0.71110**
Test 3	**0.62611**	**0.72668**	**0.71853**	**0.71853**
Test 4	**0.61367**	**0.71934**	**0.70197**	**0.70197**
Mean	**0.61439**	**0.72200**	**0.71209**	**0.71209**

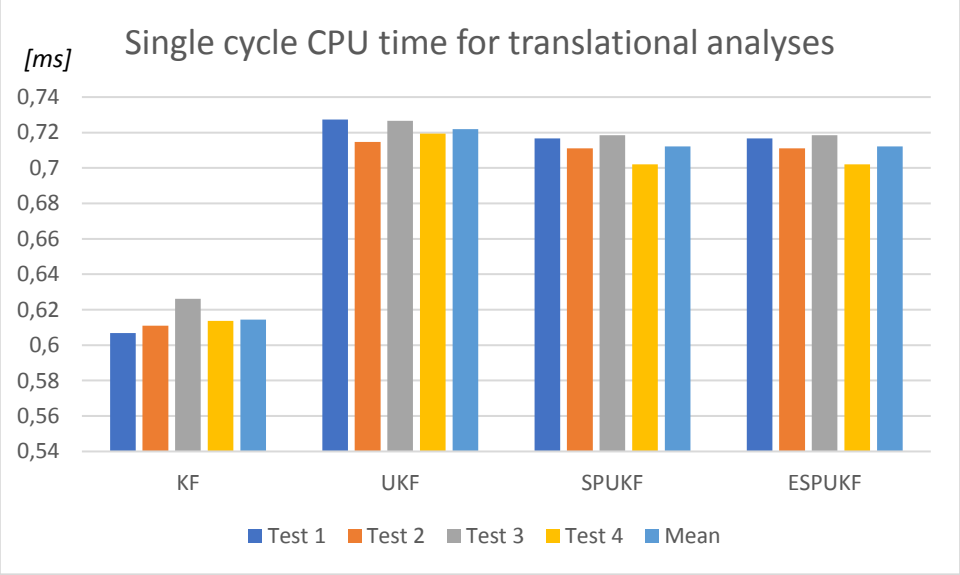

Figure 8.10: Computational times for a single cycle of the filters over all the tests and their mean.

As already seen in the ideal case, the classic Kalman filter is clearly the faster, but the Unscented filters are not much slower, since they require only one additional millisecond to purse a single cycle. It is possible to notice also that the SPUKF is still slightly faster than the UKF.

All the filters work very well for the translational motion and all can properly react to the presence of some wrong provided data. The most accurate one appears to be the SPUKF, especially for the estimation of the relative velocity and, in addition, this is faster than the basic UKF. The fastest filter is clearly the linear Kalman filter, thanks to its low complexity, that, however, leads to a lower accuracy.

8.2 Rotational motion

From the application of the non-linear filters to the relative attitude data from the monocular vision camera, we have collected the results for the mean and RMS errors between the filters' estimation and the reference's values that can be seen in Figures from 8.11 to 8.14 and in Tables from 8.8 to 8.11.

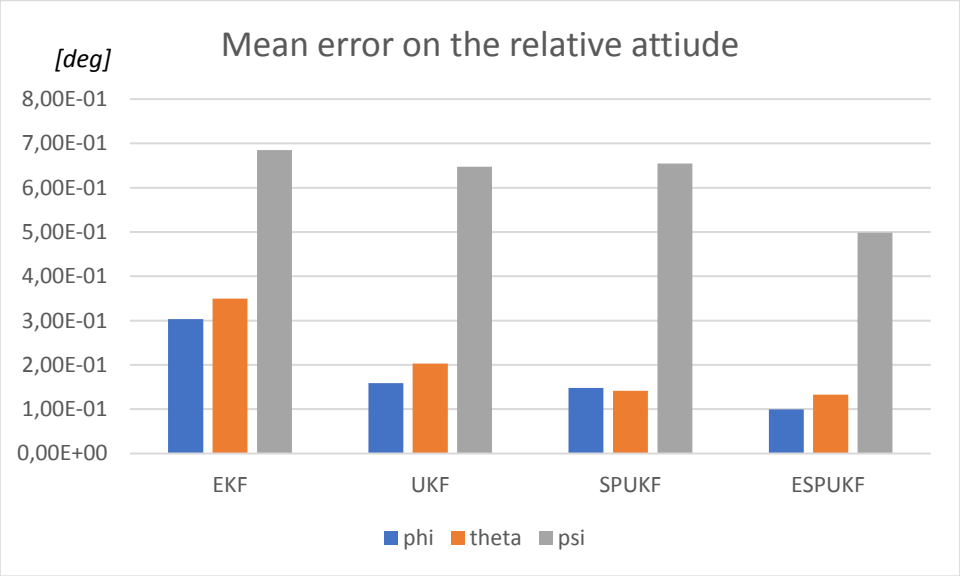

Figure 8.11: Mean errors on the relative attitude estimated by the filters, mediated over the four tests.

Table 8.8: Mean errors on the relative attitude estimated by the filters, mediated over the four tests.

Rotational Motion Monocular camera measurements		**EKF**	**UKF**	**SPUKF**	**ESPUKF**
Mean error on relative attitude	**ϕ [deg]**	**3.04E-01**	**1.59E-01**	**1.48E-01**	**9.99E-02**
	θ [deg]	**3.50E-01**	**2.03E-01**	**1.42E-01**	**1.33E-01**
	ψ [deg]	**6.85E-01**	**6.48E-01**	**6.55E-01**	**4.99E-01**

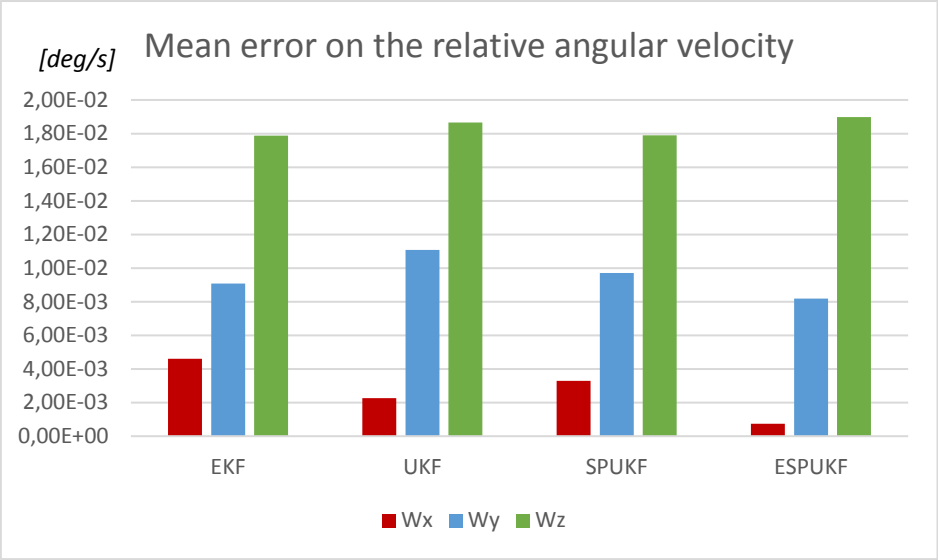

Figure 8.12: Mean errors on the relative angular velocity estimated by the filters, mediated over the four tests.

Table 8.9: Mean errors on the relative angular velocity estimated by the filters, mediated over the four tests.

Rotational Motion Monocular camera measurements		EKF	UKF	SPUKF	ESPUKF
Mean error on relative angular velocity	ω_X [deg/s]	4.60E-03	2.26E-03	3.30E-03	7.36E-04
	ω_Y [deg/s]	9.08E-03	1.11E-02	9.71E-03	8.19E-03
	ω_Z [deg/s]	1.79E-02	1.87E-02	1.79E-02	1.90E-02

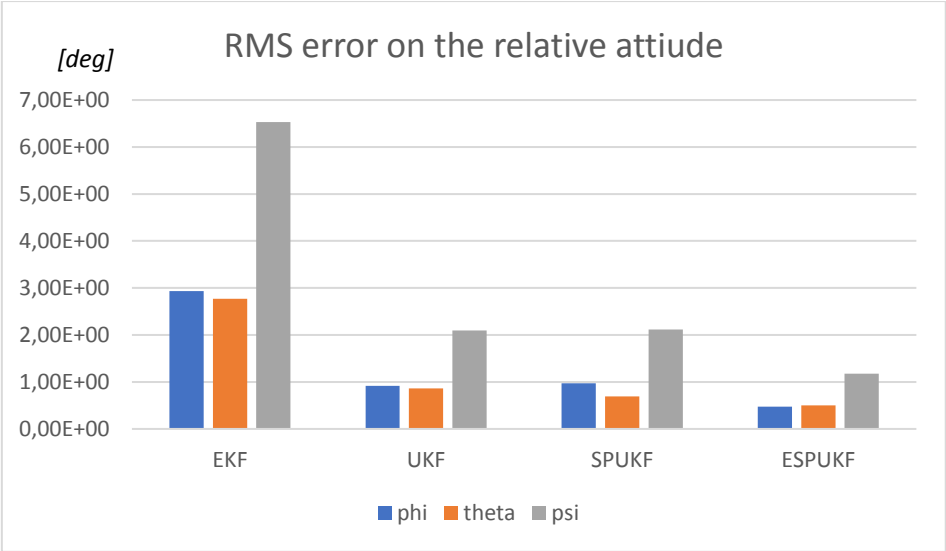

Figure 8.13: RMS errors on the relative attitude estimated by the filters, mediated over the four tests.

Table 8.10: RMS errors on the relative attitude estimated by the filters, mediated over the four tests.

Rotational Motion Monocular camera measurements		EKF	UKF	SPUKF	ESPUKF
RMS error on relative attitude	ϕ [deg]	2.93E+00	9.19E-01	9.71E-01	4.72E-01
	θ [deg]	2.77E+00	8.63E-01	6.92E-01	5.02E-01
	ψ [deg]	6.53E+00	2.10E+00	2.11E+00	1.18E+00

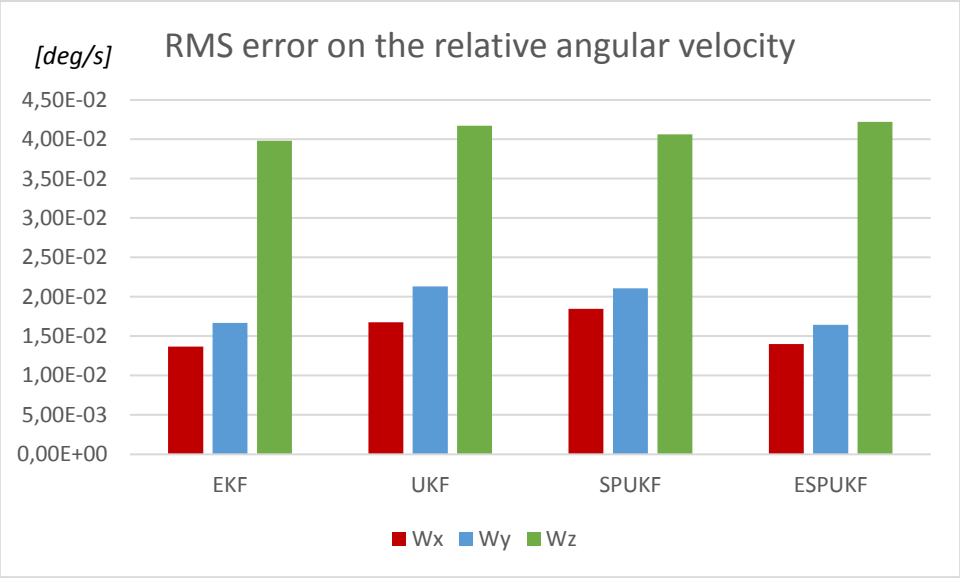

Figure 8.14: RMS errors on the relative angular velocity estimated by the filters, mediated over the four tests.

Table 8.11: RMS errors on the relative angular velocity estimated by the filters, mediated over the four tests.

Rotational Motion Monocular camera measurements		**EKF**	**UKF**	**SPUKF**	**ESPUKF**
RMS error on relative angular velocity	**ω_X [deg/s]**	**1.37E-02**	**1.68E-02**	**1.85E-02**	**1.40E-02**
	ω_Y [deg/s]	**1.67E-02**	**2.13E-02**	**2.11E-02**	**1.65E-02**
	ω_Z [deg/s]	**3.98E-02**	**4.17E-02**	**4.06E-02**	**4.22E-02**

These results show that the estimation of the relative angular velocity is similar for all the filters, while for the estimation of the relative attitude, the most accurate filter is the ESPUKF. In fact, this filter can react very well to the presence of some wrong provided data leading to the lowest mean and RMS errors for the relative attitude. Even the other two Unscented filters face quite well this problem as compared to the EKF that appears to be weaker, hence its RMS error on the relative attitude reach 6.5 degrees, while for the other filters it is around 2 degrees, or even 1 degree for the ESPUKF.

In the same way as for the other cases, the computational times of each filter were sampled and mediated obtaining the values listed in Table 8.12.

Table 8.12: Mean computational times of each filter for the analysis of the four tests.

CPU time [s]	**EKF**	**UKF**	**SPUKF**	**ESPUKF**
Test 1	**8.2146**	**9.3767**	**9.4345**	**12.571**
Test 2	**8.3815**	**9.4194**	**9.6049**	**12.715**
Test 3	**7.6893**	**8.6340**	**8.8459**	**11.798**
Test 4	**10.580**	**12.063**	**12.315**	**16.395**

The total number of prediction and update steps are the same presented before and shown in Table 8.6.

Finally, the total CPU times are divided by the sum of the two numbers of steps, the prediction and the measurements update steps, in order to evaluate the required time for a single cycle of the filters.

So, the specific computational times for a single cycle of these filters are the ones listed in Table 8.13 and displayed in Figure 8.15.

Table 8.13: Computational times for a single cycle of each filter for the analysis of the four tests and their mean.

Single cycle CPU time [ms]	EKF	UKF	SPUKF	ESPUKF
Test 1	**0.74671**	**0.85235**	**0.85760**	**1.14271**
Test 2	**0.74815**	**0.84079**	**0.85735**	**1.13496**
Test 3	**0.75719**	**0.85022**	**0.87109**	**1.16179**
Test 4	**0.74069**	**0.84451**	**0.86215**	**1.14779**
Mean	**0.74819**	**0.84697**	**0.86205**	**1.14681**

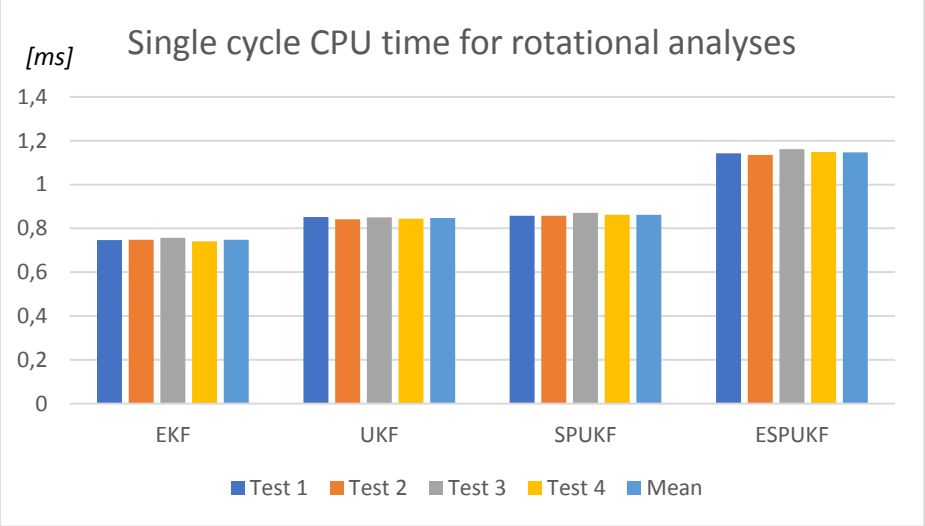

Figure 8.15: Computational times for a single cycle of the filters over all the tests and their mean.

From these final CPU times shown in Fig. 8.15, it is possible to see that the EKF is still the fastest filter, while the other three are quite slower, with the ESPUKF as the slowest. In fact, to fulfil a single cycle, it requires half a millisecond more as compared to the EKF, whereas the other two Unscented filters require only a tenth of a millisecond more.

All the Unscented filters show high accuracy even in this non-ideal condition. The most accurate and stable filter is the ESPUKF, which, however, is also the slowest in pursuing the filtering process. The other two Unscented filters are quite similar for both the accuracy and the computational time and they offer a good compromise between these two factors.
The EKF is indeed the faster filter but leads to lower precision and a weaker reaction to the presence of wrong measurements, so its employment should be associated with good and reliable measurement instruments.

Chapter 9

Sensitivity analyses

For a complete evaluation of the performances of these Unscented filters, we set up three sensitivity analyses to simulate realistic conditions taking into account the aspects described below.

- Time delay on the providing of the data, acceleration and relative position or attitude. If, for the previous analyses, the filters received immediately the collected measurements, now they are provided with some time delay, as in real conditions since the collected images have to be analyzed to estimate the relative pose between the satellites.

- Uncertainties on the moments of inertia of the target spacecraft. These uncertainties concern the evaluation of the Euler's equations within the filters. In a real case it is hardly possible to know the exact values of the moments of inertia of a target spacecraft, especially considering that these parameters can change due to expulsion of propellent or to movement of instruments and extremities.

- Noises on the provided external accelerations for both the translational and the rotational motion. Adding some noises to these accelerations it is possible to simulate the presence of a real sensor on board of the target spacecraft providing measurements of the external acceleration acting on it.

We choose to fulfill these analyses considering the measurements from the last maneuver carried out with the SPARTANS facility, since this is the more realistic one, thanks to its combined roto-translational motion.

9.1 Time delay

We evaluated the influence of the time delay in the measurements chain going from the ideal case, where there is no delay, to a delay of half a second.

The time delay on the providing of the data influences the performances of all the filters in the same way and, therefore, the evolution of the RMS errors of their outputs are almost identical. For this reason, only the trends of the RMS errors as a function of the delay of the basic Unscented Kalman filter is displayed, as it can be seen in Fig. 9.1 for translational motion and in Fig. 9.2 for rotational motion.

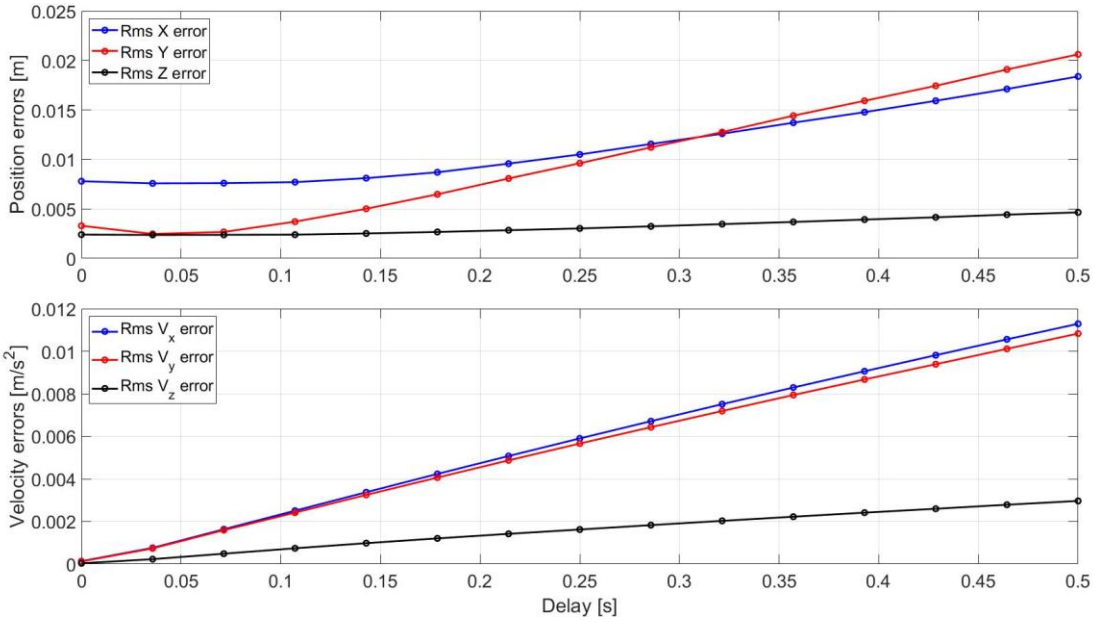

Figure 9.1: RMS errors on the translational analyses as a function of the time delay of the provided measurements.

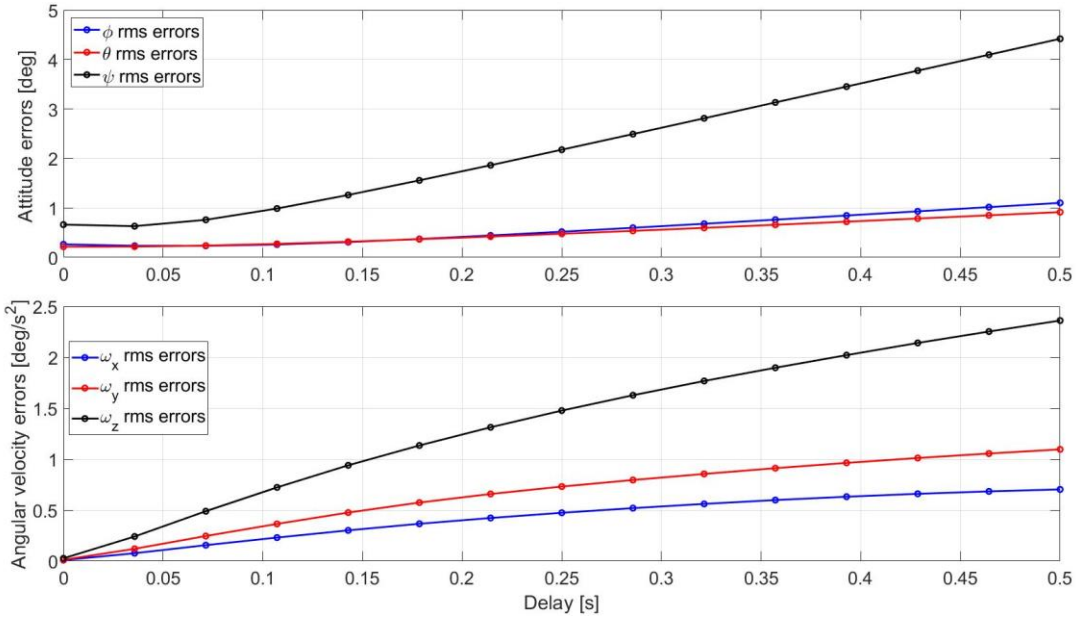

Figure 9.2: RMS errors on the rotational analyses as a function of the time delay of the provided measurements.

From these results it is possible to see that the errors due to the time delay on the providing of the measurements grow clearly along with the delay itself, but their growth depends especially on the speed of the spacecraft, either in translation or in rotation. This can be appreciated looking at which error increases more. The translation motion is characterized mainly by movements on the ***xy*** plane, because the higher velocities are $\boldsymbol{V_X}$ and $\boldsymbol{V_Y}$, and so the major errors concern these two axes. In the same way, the higher speed for the rotational motion is $\boldsymbol{\omega_Z}$ and naturally the main errors appear around the ***z*** axis.

This sensitivity analysis helps to understand that the uncertainty caused by this issue are quite contained even when the delay is at its maximum.

9.2 Uncertainties on the Moments of Inertia

To evaluate the influence, on the performances of the filters, of a not perfect knowledge of the moments of inertia of the target spacecraft we set up some analyses where we provided the filters with wrong values of the moments of inertia. These analyses refer only to the rotational motion, because in the translational motion there is not the dependence on the moments of inertia, so the comparison, in this case, concerns the EKF, the UKF, the SPUKF and the ESPUKF.

For each moment of inertia, the range of variation was set between ***± 25 %*** of its reference value. We analyzed the effects of these uncertainties singularly for each moment of inertia, then we used the obtained results to define the ranges of variation of the filters' performances as a function of the uncertainty on a general moment of inertia.

As it happened for the previous sensitivity analysis, the behavior of all the analyzed filters is very similar, so, the displayed trends of the RMS errors are those associated to the UKF (Fig. 9.3 and 9.4). In the next figures, the solid lines represent the maximum and the minimum RMS errors caused by the uncertainties on the knowledge of the moments of inertia, while the dashed lines represent the mean values of these RMS errors.

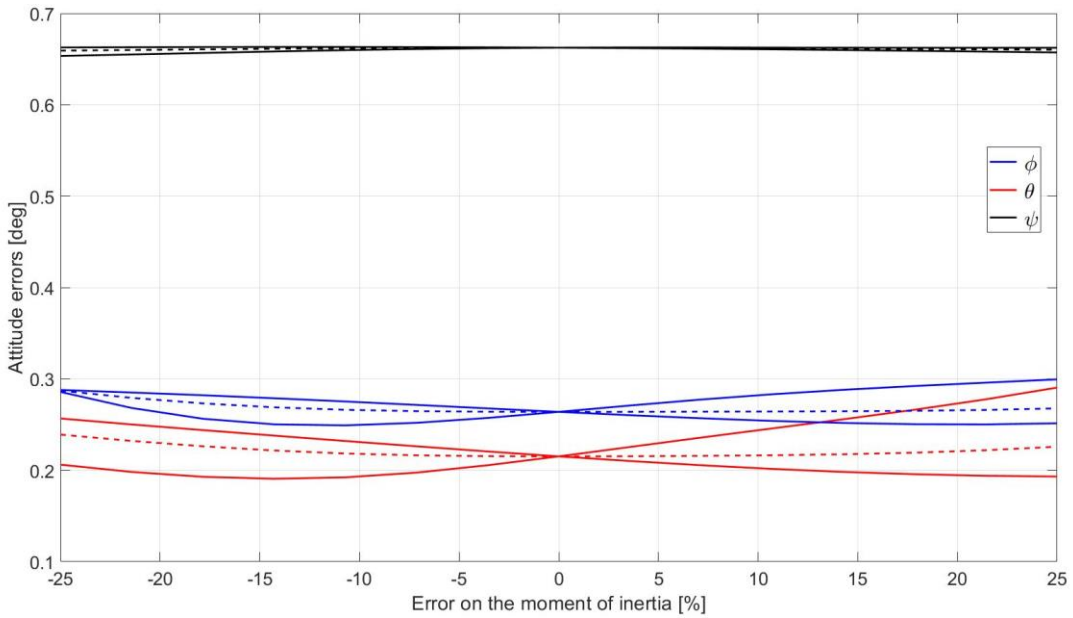

Figure 9.3: RMS errors on the attitude estimation as a function of the error on the target's moments of inertia. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

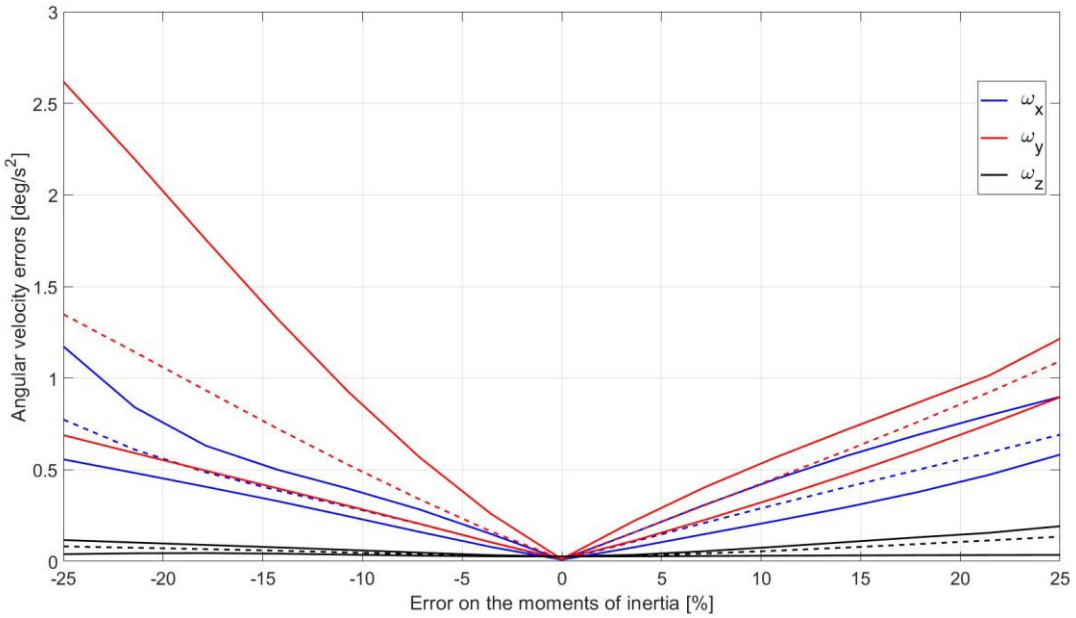

Figure 9.4: RMS errors on the relative angular velocity estimation as a function of the error on the moments of inertia. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

It is easy to notice that these uncertainties do not influence very much the accuracy of the attitude estimation of the filters (Fig. 9.3). Considering the errors on the estimation of the angular velocity (Fig. 9.4), the influence of the uncertainties is stronger, but, analyzing a range restricted to ***± 10 %*** of the reference value of the moments of inertia, typical of various moments of inertia estimator, the errors are limited within a maximum of ***1 deg/s***.

The effects of these uncertainties are small, because the moments of inertia are used only on the Euler's equations during the propagation of the angular velocity. So, they concern only a component of variation of the angular velocity that is contained if compared to the values of the external accelerations.
On the other hand, even if the effects on the angular velocity are contained, they induce the filters to diverge from the correct values of the relative attitude. However, the estimation of the attitude in the filters still relies on the attitude measurements provided. For this reason, the filters manage to oppose to the divergence induced and their performances on the attitude estimation are kept stable.

9.3 Noises on the external accelerations

Finally, we evaluated the performances of the filters when the provided external accelerations are characterized by some noises. To do so, we added errors with a random linear distribution over a maximum range of $\pm 0.1\ m/s^2$ for the translational analyses, and of $\pm 5\ deg/s^2$ for the rotational analyses.

We did not developed the full formulation of the errors of a real sensor, because it involved the addition of some parameters and equations to the filters and the focus of this sensitivity analysis is the evaluation of the performances of the filters to these kind of errors and not their specific set up in this case.

To evaluate the influence of the noises on the external accelerations we set up several analyses, starting from the ideal condition and adding progressively noises until the maximum ranges defined, realizing ten steps for the values of the noises' ranges. For each of these steps we created randomly thirty times the noisy external accelerations. This results in 300 analyses for each of the two motions and we collected the RMS errors on the state parameters from all the filters. The images below display the mean, the maximum and the minimum of the RMS errors obtained for every state parameter as a function of the range of the added noise.

Differently from the two previous studies, here there are valuable difference between the performances of the classic Kalman filters and the Unscented ones. So, the following images concern the output of the KF, the UKF and the SPUKF, for the translational motion, and, for the rotational motion, the EKF and the UKF, since the other two Unscented filters show the same performances of the UKF.

For the translational motion the two last filters, the SPUKF and the ESPUKF, are identical, so we refer only to the SPUKF for their evaluation.

As for the graphs of the previous analysis, in the next figures (from Fig. 9.5 to 9.14) the dashed line refers to the mean values of the RMS errors and the solid lines refer to the limits of the range of variation of the RMS errors.

9.3.1 Translational motion

The influence of the acceleration noise on the accuracy of each filter for the relative position estimation can be seen in Figures from 9.5 to 9.7.

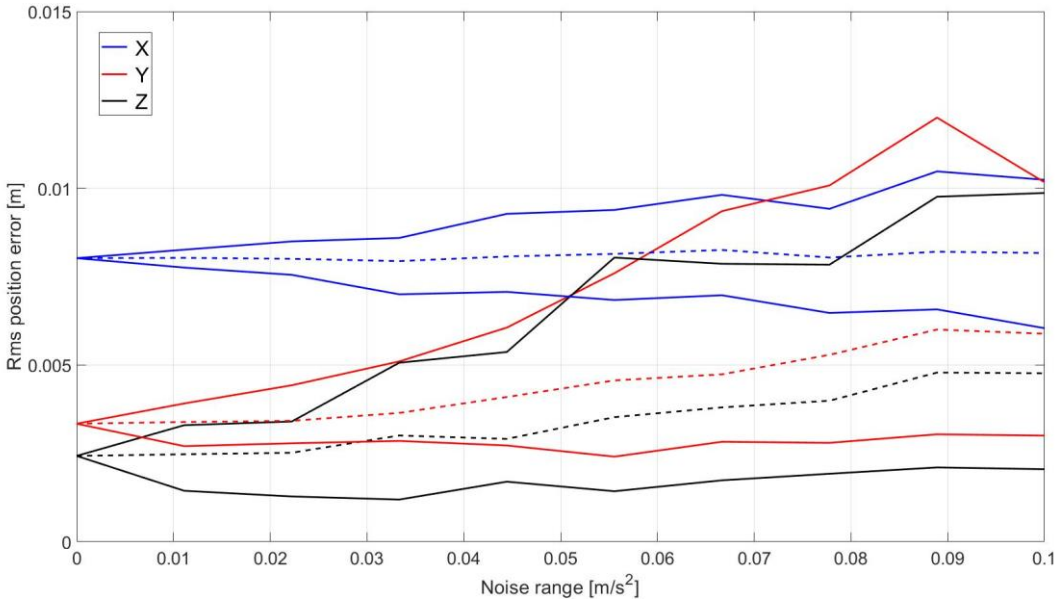

Figure 9.5: Trends of the mean RMS errors and their ranges for the linear Kalman filter. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

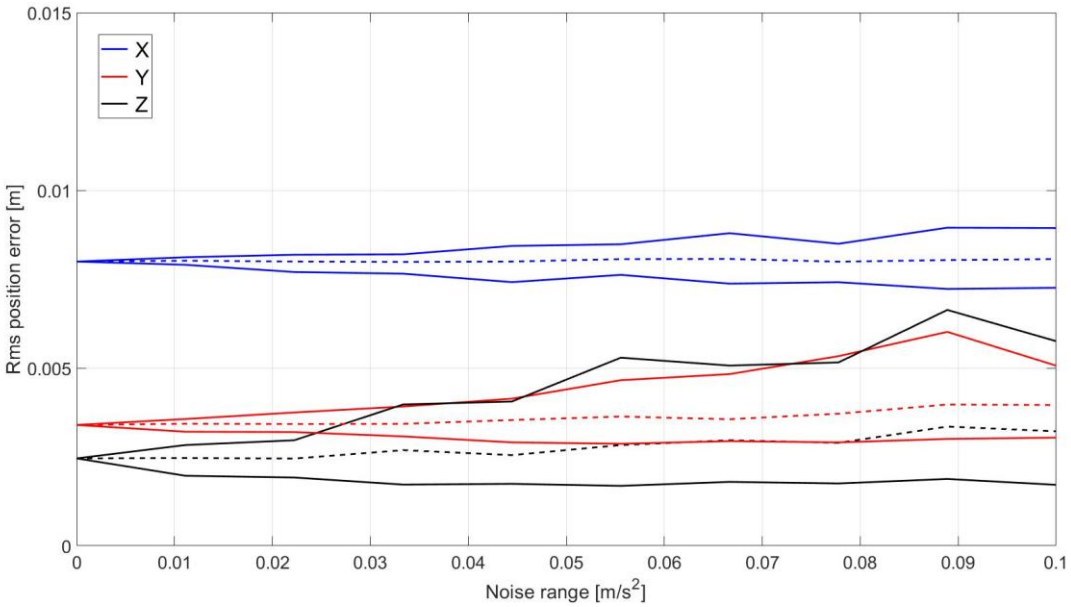

Figure 9.6: Trends of the mean RMS errors and their ranges for the Unscented Kalman filter. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

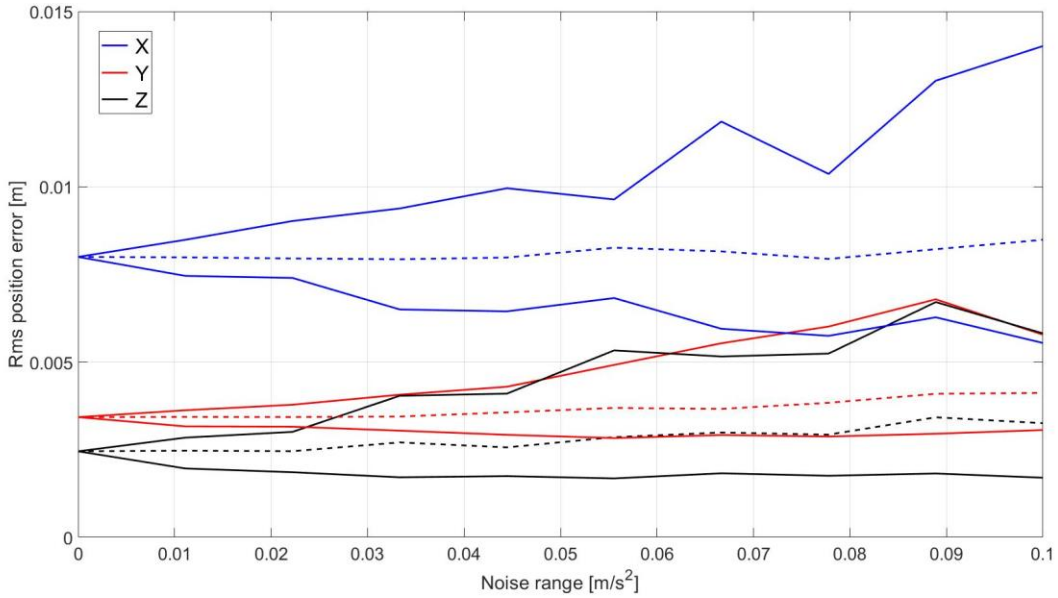

Figure 9.7: Trends of the mean RMS errors and their ranges for the SPUKF. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

It is possible to notice that the Unscented Kalman filter shows higher estimation accuracy even when the external accelerations are noisy, since the mean RMS errors and their ranges of variation are smaller and more stable with respect to the ones associated to the linear Kalman filter. The other Unscented filter, the SPUKF, shows a different behavior, since it leads to small errors for the ***y*** and the ***z*** coordinates, as for the UKF, while for the ***x*** coordinate leads to the highest errors, even higher than the one associated to the KF. However, the performances of all the filters are good even when the acceleration noise is high, so all the filters have good robustness to this uncertainty.

From Fig. 9.8, 9.9 and 9.10 it is possible to notice the influence of the acceleration's noise on the performances for the estimation of the relative translational velocity.

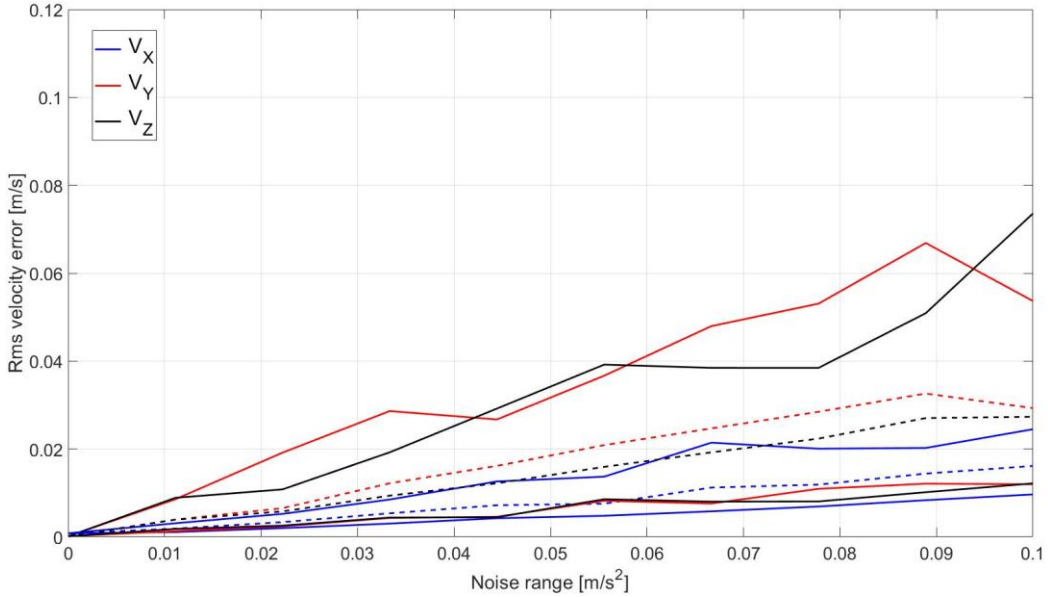

Figure 9.8: Trends of the mean RMS errors and their ranges for the linear Kalman filter. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

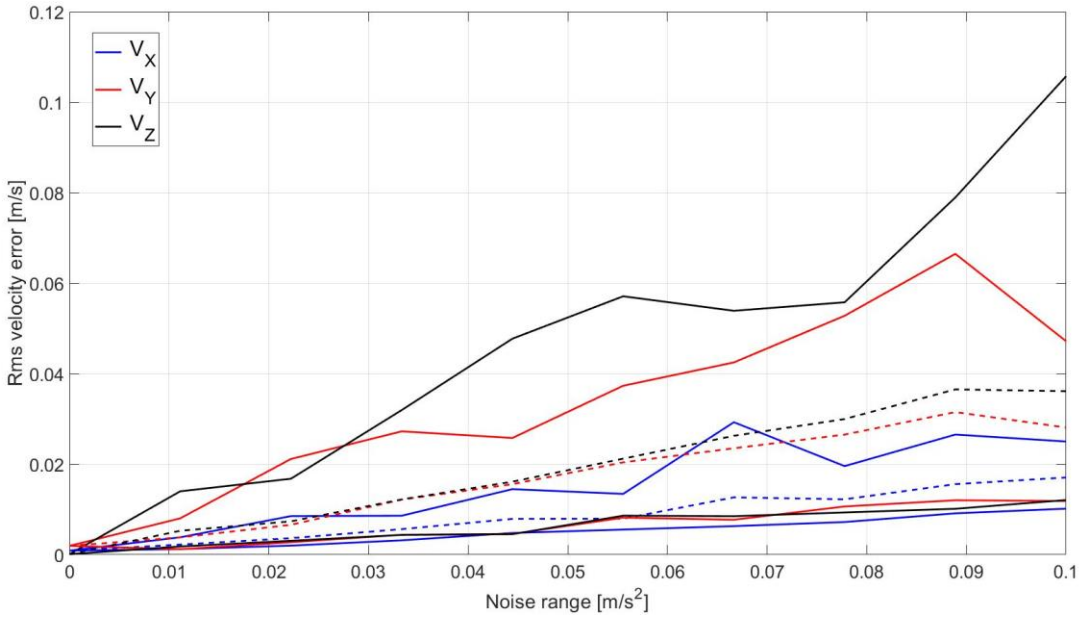

Figure 9.9: Trends of the mean RMS errors and their ranges for the Unscented Kalman filter. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

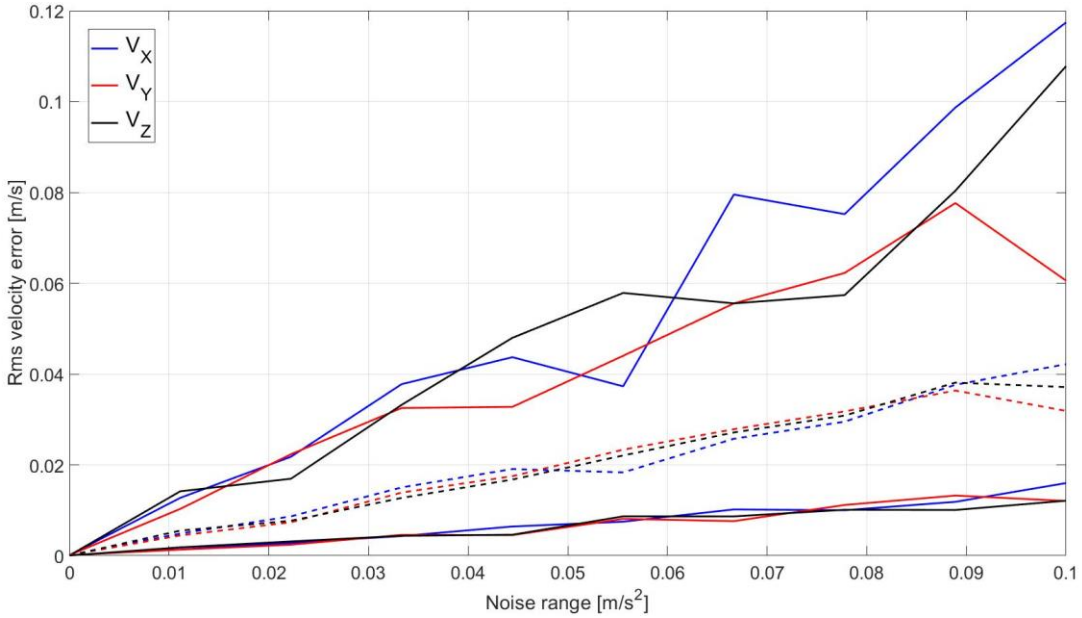

Figure 9.10: Trends of the mean RMS errors and their ranges for the SPUKF. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

Comparing these images (Fig. 9.8, 9.9 and 9.10), it is possible to see that the behaviors of the first two filters are similar when considering their errors on the relative velocity. The only valuable difference concerns the errors on the velocity along the ***z*** axes that are slightly higher for the UKF.
On the other hand, the SPUKF appears the more affected one. In fact, it is characterized by the higher values of the errors for all the three velocities.

Anyway, the developed errors are small for all the filters that maintain good performances.

9.3.2 Rotational motion

As already said, for this motion, the three Unscented filters show the same performances in response to the presence of noise on the external angular acceleration, so only the UKF trends are displayed.

The influence of the acceleration noise on the accuracy of the filters for the relative attitude estimation can be seen in the Figures 9.11 and 9.12.

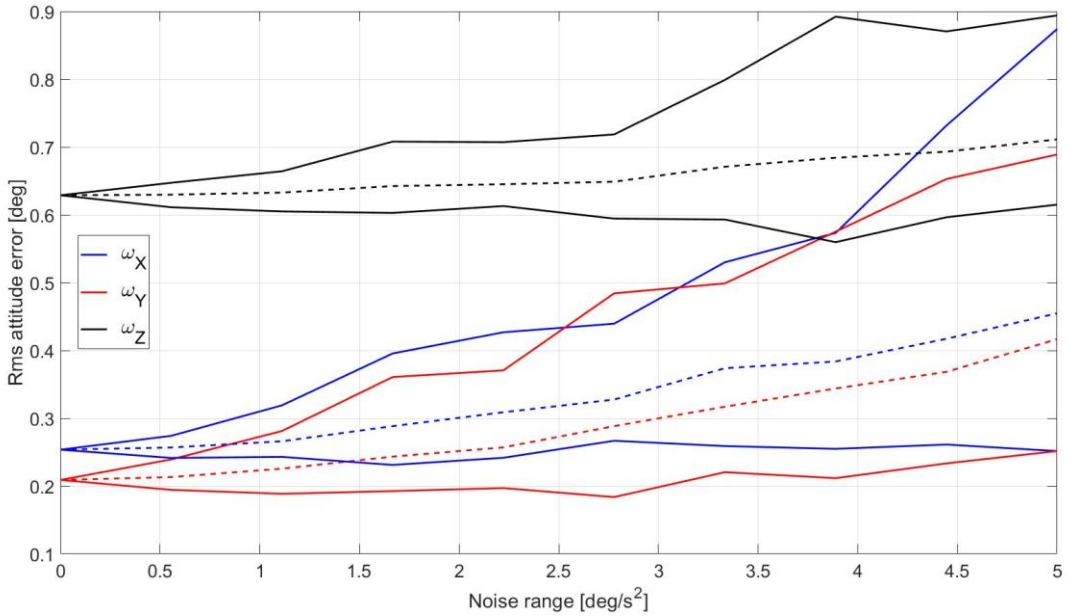

Figure 9.11: Trends of the mean RMS errors and their ranges for the EKF. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

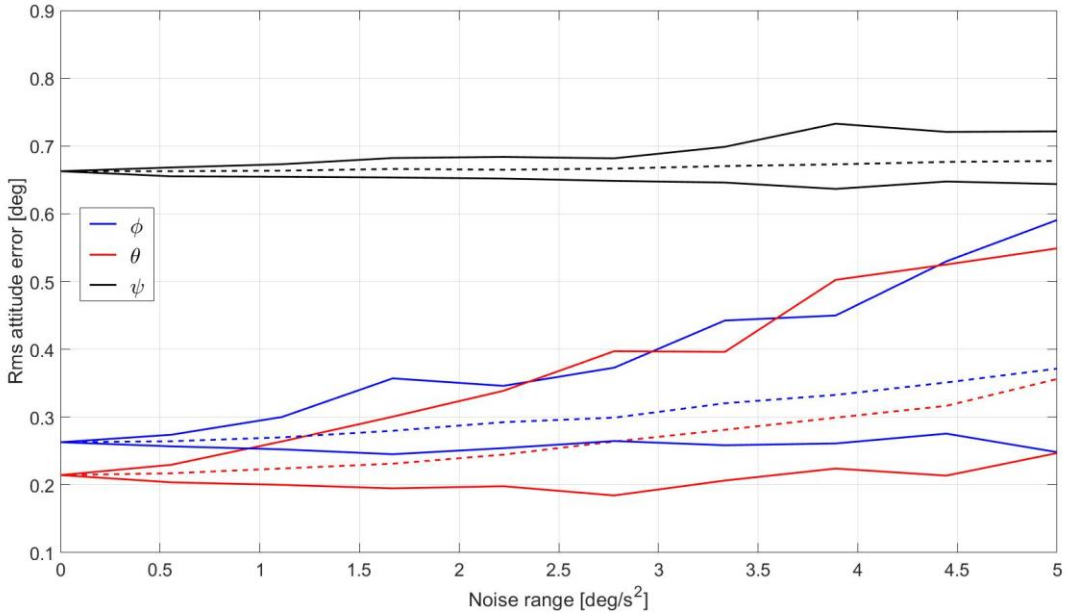

Figure 9.12: Trends of the mean RMS errors and their ranges for the UKF. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

Even here it is possible to see that the Unscented filters are less affected by the presence of noises on the external angular accelerations provided. In fact, both the mean value of the RMS errors and their ranges of variation are lower for these filters with respect to the Extended Kalman filter.
However, all the filters maintain good performances in these conditions, without diverging to great errors. This thanks to the providing of attitude measurements that prevent the drift of the estimation of the filters. From the next Fig. 9.13 and 9.14 it is possible to notice the influence of the acceleration's noise on the performances for the estimation of the relative angular velocity.

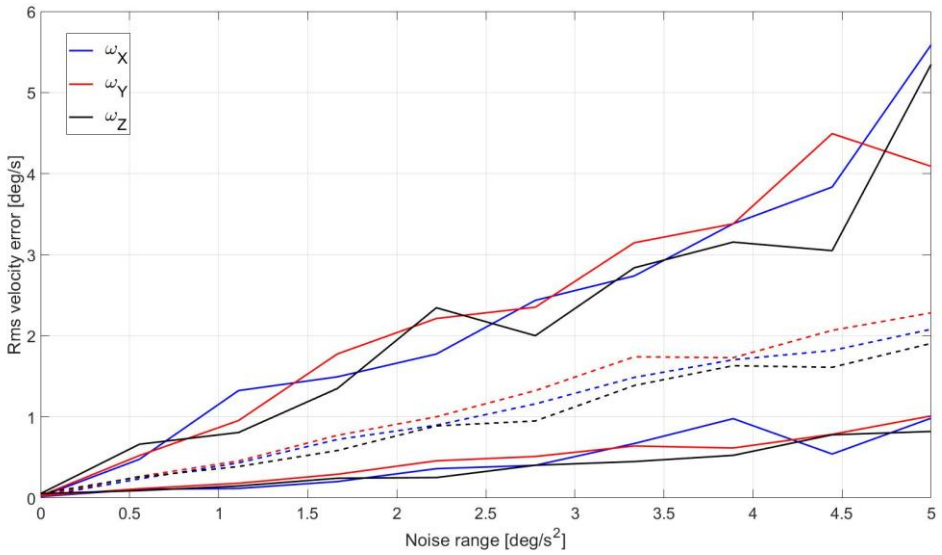

Figure 9.13: Trends of the mean RMS errors and their ranges for the EKF. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

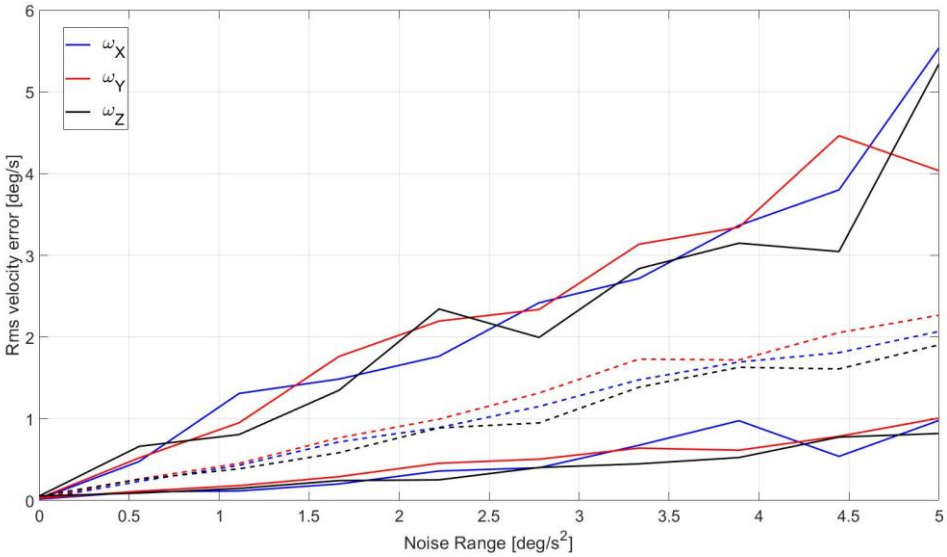

Figure 9.14: Trends of the mean RMS errors and their ranges for the UKF. The solid lines represent the maximum and the minimum of the RMS errors' ranges of variation while the dashed lines represent the mean values of the RMS errors.

Differently from before, for the translation motion, looking at the errors on the estimation of the relative angular velocity, valuable differences do not come up and the behaviors of the filters appear similar. It is important to notice that the developed errors are contained also for these parameters, since the mean RMS errors of the angular velocities reach only ***2 deg/s*** when the accelerations' noises are at their maximum.

Chapter 10
Conclusions

The main goal of this thesis was to propose new alternatives for the navigation estimator filters currently employed for close proximity space missions. Currently, the most widely used filters for this purpose are the linear Kalman Filter (KF) and the Extended Kalman Filter (EKF), respectively for the estimation of the relative translational motion and for the estimation of the relative rotational motion. In this thesis three alternative formulations to the standard Kalman filters are presented: the Unscented Kalman Filter (UKF) and two other variants of this, i.e. the Single Propagation Unscented Kalman Filter (SPUKF) and the Extrapolated Single Propagation Unscented Kalman Filter (ESPUKF).

The activities pursued to fulfil the aim of this thesis are summarized as follows.

- Collection of experimental relative dynamics measurements employing the SPARTANS facility of the University of Padova.

- Determination of all the parameters necessary for the correct set-up of the filters, such as the moments of inertia of the target spacecraft and the variances of the collected measurements.

- Development and set up of all the filters for the relative dynamics estimation splitting the analyses of the translational motion and the rotational one. This ends up on the formulation of eight filters: four linear filters for the translational dynamics, the KF and the UKF, the SPUKF and the ESPUKF in their linear form; and other four, non-linear, for the rotational dynamics, the EKF and again the UKF, the SPUKF and the ESPUKF this time in their classic non-linear form.

- Test and comparison of these developed filters with numerical simulations over the experimental measurements collected.

- Evaluation and comparison of the performances of the filters for more realistic conditions, carrying out sensitivity analyses on the influences of: time delay on the availability of the measurements provided by the MC system and the monocular camera on board the chaser satellite; uncertainties on the moments of inertia of the target; noise on the external accelerations.

The results of the numerical analyses carried out on experimental data showed that the Unscented Kalman filters analyzed in this thesis appeared as valuable alternatives to the standard ones, since they lead to high estimation accuracy for both translational and rotational motions. In addition, in order to ensure high accuracy level these filters introduce more complexity to the navigation algorithm and along with the increase of the complexity there is naturally an increase of the computational time.

Following these considerations, the developed Unscented filters offers better estimation accuracy with longer computational time with respect to the classic KF and the EKF considering respectively the translational and the rotational dynamics.

The translational motion is linear and simple and so all the filters show good estimation accuracy. The filters performances are similar in this case, especially for the relative position estimation since all the RMS errors of the filters stand below **1 cm** with a baseline of **2 m**. Considering the relative velocity estimation, it is possible to notice the better accuracy of the Unscented filters, even though the RMS errors associated to the KF are already contained under **1 mm/s** with a baseline of almost **200 mm/s**.
Even the computational times are similar in this case, but the KF is indeed the fastest since it is associated to a single cycle CPU time lower than **0.62 ms** while the single cycle CPU time of the Unscented filters stands around **0.72 ms**.

On the other hand, the rotational motion is non-linear and more complex and so the differences between the performances of the filters become more evident, especially for the relative attitude estimation, since the accuracy on the estimation of the relative angular velocity is similar for all the filters. Considering the relative attitude estimation, it is possible to notice the improvement brought by the Unscented filters, since the RMS errors goes from a maximum of **6.53 deg** associated to the EKF to a value of **1.18 deg** associated to the ESPUKF.
The computational times differ more in this rotational motion case, going from **0.75 ms** for the EKF up to **1.15 ms** for the ESPUKF.

Among the three Unscented filters the comparison of their performances is presented as follows.

- For the translational dynamics estimation, the last two filters, the SPUKF and the ESPUKF, become identical after their set-up and so the considerations are referred only to the SPUKF.
  Comparing the performances of the UKF and the SPUKF it is possible to notice that they share the same level of accuracy for the relative position estimation, while the second filter leads to better accuracy for the relative velocity estimation by almost an order of magnitude, since its major RMS error is $\mathbf{1.04 \cdot 10^{-4}\ m/s}$, while the major error of the UKF is $\mathbf{4.9 \cdot 10^{-4}\ m/s}$. In addition, the SPUKF appears slightly faster, with a single cycle CPU time of **0.717 ms** with respect to the **0.727 ms** of the UKF. For this reason, the most valuable alternative for the translational dynamics is the SPUKF.

- For the rotational dynamics, the last two filters are not identical and can be evaluated singularly. All the three Unscented filters offer diverse good compromises between estimation accuracy and CPU time. Going from the UKF to the ESPUKF it is possible to notice an increase in both the estimation accuracy and the CPU time. So, the more accurate is the desired estimation, the more the computational time will be high.
  The performances of the filters for the estimation of the relative angular velocity are similar, since they all lead to maximum RMS errors below **0.05 deg/s**. Considering the accuracy on the relative attitude estimation, it is possible to notice that the ESPUKF leads to better accuracy than the other two filters that present similar performances. Between the RMS errors of the ESPUKF and the ones of the UKF there is almost a factor 2 since, for example, their major errors are **1.18 deg** for the ESPUKF and **2.11** for the UKF. The benefits on the estimation accuracy, however, imply a higher computational time. Indeed, the ESPUKF is associated to a single cycle CPU time of **1.15 ms**, while the other two filters are associated to a single cycle CPU time of **0.85 ms**.

From the sensitivity analyses appeared that all the filters are affected in the same way by the time delay on the measurements input and by the uncertainties on the moments of inertia of the target spacecraft.
Considering the noise on the external accelerations, the standard filters appeared more affected than the Unscented filters, especially for the UKF, which behaves better for both motions.

Overall, with respect to all these realistic conditions all the filters appeared robust maintaining good estimation accuracy since, over all these analyses, the maximum RMS errors that come up are: **0.02 m** and **0.12 m/**s for the relative position and velocity estimation and **4 deg** and **5 deg/s** for the relative attitude and angular velocity estimation.

To conclude, the developed Unscented filters results as valuable alternatives to the standard filters currently used, offering the possibility to choose which filter best fits the mission requirement. Depending on the accuracy of the vision camera used to estimate the relative pose between the two spacecraft and on the computational capacity of the on-board computer, it is possible to select the filter that best combines the characteristic of these two systems; e.g. the unscented filters are the best choice for spacecraft hosting low accuracy camera but advanced on-board computer.

Bibliography

[1] A. Valmorbida, *Development and Testing of Model Predictive Control Strategies for Spacecraft Formation Flying,* Università degli studi di Padova, Scuola di Dottorato di Ricerca in: Scienze Tecnologiche e Misure Spaziali, PhD. Thesis.

[2] M. Mazzuccato, *Controllo d'assetto a tre assi di un simulatore per il volo in formazione – Software Design e Hardware Test,* Università degli studi di Padova, Master Degree Thesis.

[3] M. Mazzuccato, *Design and Testing of a Vision Based Navigation System for a Spacecraft Formation Flying Simulator*, Università degli studi di Padova, Scuola di Dottorato di Ricerca in: Scienze Tecnologiche e Misure Spaziali, PhD. Thesis.

[4] A. Valmorbida, M. Mazzuccato and M. Pertile, *Calibration procedures of a vision-based system for relative motion estimation between satellites flying in proximity,* Measurement, 2019, https://doi.org/10.1016/j.measurement.2019.107161 .

[5] A. Valmorbida, M. Mazzuccato, S. Tronco, M. Pertile, E. C. Lorenzini, *Design of a ground-based facility to reproduce satellite relative motions*, 2017 IEEE Metrology for Aerospace (MetroAeroSpace), DOI: 10.1109/MetroAeroSpace.2017.7999619.

[6] M. Mazzuccato, A. Valmorbida, A. Guzzo, E. C. Lorenzini, *Stereoscopic vision-based relative navigation for spacecraft proximity operations*, 2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), DOI: 10.1109/MetroAeroSpace.2018.8453524.

[7] A. Valmorbida, M. Mazzucato, S. Tronco, S. Debei, E. C. Lorenzini, *SPARTANS - A cooperating spacecraft testbed for autonomous proximity operations experiments*, Instrumentation and Measurement Technology Conference (I2MTC), 2015 IEEE International, DOI: 10.1109/I2MTC.2015.7151360.

[8] V. Pesce, R Opromolla, S. Sarno, M. Lavagna, M. Grassi, *Autonomous relative navigation around uncooperative spacecraft based on a single cameraI*, Aerospace Science and Technology 84 (2019) 1070–1080.

[9] V. Pesce, M. Farooq Haydar, M. Lavagna, M. Lovera, *Comparison of filtering techniques for relative attitude estimation of uncooperative space objects*, Aerospace Science and Technology 84 (2019) 318–328.

[10] V. Pesce, M. Lavagna, R. Bevilacqua, *Stereovision-based pose and inertia estimation of unknown and uncooperative space objects*, Advances in Space Research 59 (2017) 236–251.

[11] S. K. Biswas, L. Qiao, A. G. Dempster, *A Novel a Priori State Computation Strategy for the Unscented Kalman Filter to Improve Computational Efficiency*, IEEE Transactions on Automatic Control, Vol.62, n° 4, April 2017.

[12] S. K. Biswas, B. Southwell, A. G. Dempster, *Performance analysis of Fast Unscented Kalman Filters for Attitude Determination*, IFAC PapersOnLine 51-1 (2018) 697–701.

[13] S. K. Biswas, L. Qiao, A. G. Dempster, *Computationally Efficient Unscented Kalman Filtering Techniques for Launch Vehicle Navigation using a Space-borne GPS Receiver*, ION GNSS+, 2016.

[14] Jason L. Anderson, Dr. Franz J. Kurfess, Dr. Jordi Puig-Suari, *A framework for developing artificial intelligence for autonomous satellite operations*, IJCAI–09 Workshop on Artificial Intelligence in Space, Pasadena, California, US, 2009.

[15] ZED camera website, http://www.stereolabs.com .

[16] DUO camera website, http://www.codelaboratories.com .

[17] Accelerometers and gyros information, https://www.arrow.com/en/categories/sensors .