Politecnico di Milano

Department of Aerospace Science and Technology

Master Course in Space Engineering

High Order Filters for Relative Pose Estimation of an Uncooperative Target

Advisor:
Prof. Pierluigi Di Lizia

Co-Advisors:
Prof. Mauro Massari
Dr. Francesco Cavenago

Candidate:
Simone Servadio
# 852400

Academic Year 2016-2017

Abstract

The sequential state estimation of the relative pose between a chaser and its target becomes a complex problem when the dynamics of the satellites and the measurements equation are highly nonlinear. The classical Extended Kalman Filter may show difficulties to filter the state of the target, thus high order filters that are able to capture mean and covariance more accurately has been developed in this work. This thesis explains how the first two moments, i.e. mean and covariance, are propagated with different techniques such as Differential Algebra and the Unscented Transformation, and highlights their improvements compared to linearization. These new techniques are used as building blocks for three new kind of filters: the DA-based high order extended Kalman Filter, the Unscented Kalman Filter, and the DA-based Unscented Kalman Filter. The performances of the resulting filters are assessed and compared on a test case, addressing the problem of proximity operations around the target ENVISAT. After describing the satellite dynamics, a detailed discussion on the measurements model is presented. Markers positions are the typical data obtained from the image processing software of a camera. Thus, their relation with the relative state and their visibility are discussed. Then, a Monte Carlo analysis is reported to show the improvements of the presented filters with respect to the classical implementation of the extended Kalman filter in terms of mean error and computational time. In addition, the sensitivity of the performance to the number and distribution of the markers is investigated, and an analysis of the consequences of the lack of measurements is carried out to prove robustness.

Abstract

La stima sequenziale dello stato della posa relativa tra un inseguitore ed il suo target diventa un problema complicato quando la dinamica dei satelliti e l'equazione delle misure presenta alte nonlinearitá. Il classico filtro di Kalman esteso puó presentare difficoltá nel filtrare lo stato del target, di conseguenza sono stati sviluppati filtri di alto ordine capaci di identificare la media e la covarianza in maniera piú accurata in questo elaborato. Questa tesi espone come i primi due momenti, i.e. media e covarianza, vengano propagati da diverse tecniche quali l'Algebra Differenziale (DA) e l' Unscented Transformation, e sottolinea i loro miglioramenti rispetto alla linearizzazione. Queste nuove tecniche sono i blocchi fondamentali di tre nuove tipologie di filtri: il DA-based high order extended Kalman Filter (filtro di Kalman esteso di alto ordine basato sulla DA), l' Unscented Kalman Filter, e il DA-based Unscented Kalman Filter. Le prestazioni dei filtri sono valutate e paragonate su un test case, indirizzandosi al problema delle operazioni di avvicinamento al target ENVISAT. Dopo aver descritto la dinamica del satellite, viene presentata una dettagliata discussione sul modello di misura. Le posizioni dei markers sono la tipica informazione che viene ricevuta dal software che processa le immagini di una camera. Quindi, la loro relazione con lo stato relativo e la loro visibilitá deve essere discussa. Successivamente, una analisi Monte Carlo viene riportata per mostrare i miglioramenti dei filtri descritti rispetto alla classica implementazione del filtro di Kalman esteso in termini di errore medio e tempo computazionale. Inoltre, la sensitivitá delle prestazioni rispetto al numero e alla distribuzione dei markers é investigata, e una analisi sulle conseguenze della mancanza di misure é effettuata per dimostrare robustezza.

Acknowledgements

The first “thank” goes to my parents that supported me during all these college years and did everything they could to let me focus on my studies.

A special “thank” to my advisor Professor Pierluigi Di Lizia, whose guidance, patience and knowledge helped me work in harmony and serenity.

I want to show my gratitude also to Professor Mauro Massari for all his precious advices: thanks to him I really enjoyed my last college semester despite the big amount of work.

Another “thank” goes to the PhD Candidate Francesco Cavenago. I really thank him for sharing its precious work with me and for having joined interest in my thesis.

Finally a “thank” to my friends, who helped in the difficult task of distract me from my work, and to my girlfriend, who has been always by my side and had to stand me during my days of coding.

Contents

List of Figures

List of Tables

Acronym	**Meaning**
ADR	Active Debris Removal
DA	Differential Algebra
DACE	Differential Algebra Computational Engine
DOF	Degree Of Freedom
EKF	Extended Kalman Filter
EKFDA	DA-based Extended Kalman Filter
FP	Floating Point
GNC	Guidance, Navigation and Control
GRV	Gaussian Random Variable
KF	Kalman Filter
MC	Monte Carlo
MRP	Modified Rodriguez Parameters
ODE	Ordinary Differential Equation
RMSE	Root Mean Square Error
UT	Unscented Transformation
UKF	Unscented Kalman Filter
UKFDA	DA-based Unscented Kalman Filter

Table 1: Acronyms

CHAPTER 1

Introduction

The problem of nonlinear uncertainty propagation represents a crucial issue in celestial mechanics since all practical systems, from the basic vehicle navigation to orbit determination, and mostly target tracking, involve marked non-linearities of one kind or another. Therefore, the development of suitable sequential state estimation algorithms that improve the classical Kalman filter is a problem that well answers nowadays space applications requests [42]. Different techniques have been developed and this thesis focuses deeply on two of them: the introduction of high order moments in the estimation process using Differential Algebra (DA) techniques; and the implementation of numerically efficient Unscented Transformation (UT) in Unscented Kalman filters. These filters find applications in space surveillance and spacecraft navigation since they require methods to estimate the state of orbiting objects including uncertainty to propagate and track them. Within this problems, nonlinearities arises even in simple operations, such as conversions between different coordinate systems, e.g. the conversion from polar to Cartesian coordinates [22] [23] that forms the foundation for the observation models of many sensors. The nonlinearity of such transformations entails the problem of managing the nonlinear propagation of the estimated statistics to the estimated statistics.

Several approximated techniques exist in literature to approximate the initial condition uncertainty evolution, which is typically required within any state estimation algorithm. Present-day approaches mainly rely on linearized propagation models, such as in the extended Kalman filter, or full nonlinear Monte Carlo simulations. Both solutions have their drawbacks: on one hand, the linear assumption could simplify the problem too radically leading to the drop of the solution accuracy in the case of highly marked nonlinear systems and/or long time propagations. On the other hand, Monte Carlo simulations provide true trajectory statistics but they are computationally very expensive. Therefore, Differential Algebra is introduced in the sequential state estimation problem to overcome the limits of the simple linearization by adding high order moments in the prediction step. In this way, the DA-based filter better handles those systems with highly marked non-linearities thanks to the introduction of higher order information. In fact, DA enables the possibility to compute and use the Taylor expansion up to an arbitrary order, including then nonlinearities in the filtering process.

The unscented transformation (UT) was developed to address the deficiencies of linearization as well: a set of samples are deterministically chosen to match the mean and covariance of a (non necessarily Gaussian-distributed) probability distribution. The UT is founded on the idea that it is easier to approximate a probability density function than it is to approximate an arbitrary nonlinear function or transformation [20] [23]. The DA is then introduced in the UT leading to the creation of the DA-based unscented Kalman filter (UKFDA) which can achieve better performances with respect to the basic UKF. The ability of DA to map the nonlinear propagation of uncertainty can reduce the computational cost of the UT: it is sufficient to propagate only one sample point (usually the mean) and not the whole set to accurately describe the probability distribution.

1.1 Filtering

The concept of filtering is regularly stated when the time at which an estimate is desired coincides with the last measurement points. “Sequential state estimation” and “filtering” are used synonymously throughout the remainder of the thesis: sequential state estimation is often used not only to reconstruct state variables but also to “filter” out the noisy measurement process [11]. Thus, “sequential state estimation” and “filtering” are often interchanged in the literature.

An estimation filter provides a rigorous theoretical approach to estimate the state of a filter based upon stochastic processes for the measurement error and error model. There is no knowledge of the exact values of these errors; however, there is an assumption on the nature of the errors, indeed a zero-mean Gaussian noise process is often assumed. The nonlinear filtering problem has got a fundamental role in many space-related applications such as space navigation problems and orbit determination. As pointed out before, relative pose estimation for rendezvous maneuvers with uncooperative targets demands GNC systems with an accurate encoded filter technique which must be able to perform an accurate trajectory estimation in a very reduced lapse of time.

Probably the most widely used estimator for nonlinear systems is the extended Kalman filter (EKF). The EKF applies the Kalman filter to nonlinear systems by simply linearising all the nonlinear models so that the traditional linear Kalman filter equations can be applied. Assume the equations of motion and the measurement equations are

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k), k] + \mathbf{v}(k) \tag{1.1}$$
$$\mathbf{z}(k+1) = \mathbf{h}[\mathbf{x}(k), \mathbf{u}(k), k] + \mathbf{w}(k) \tag{1.2}$$

where $\mathbf{f}[\cdot,\cdot,\cdot]$ is the process model, $\mathbf{x}(k)$ is the state of the spacecraft at time-step k, $\mathbf{u}(k+1)$ is the input vector and $\mathbf{v}(k)$ is the process noise perturbing the spacecraft state, $\mathbf{z}(k+1)$ is the actual measurement at time-step $k+1$, $\mathbf{h}[\cdot,\cdot,\cdot]$ is the measurement function, and $\mathbf{w}(k)$ is the measurement noise characterizing the observation error. These equations get linearized via Taylor expansion series around the current mean and covariance. Starting from the first two moments (i.e. mean and covariance) of the state vector at time step k, the EKF follows a division in two main steps, which are shown in Figure 1.1.

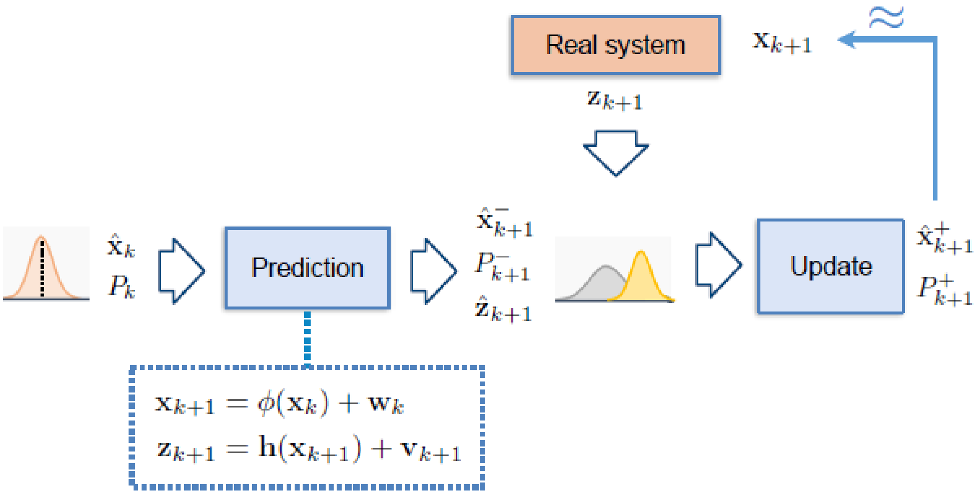
Real system
$\mathbf{x}_{k+1}$
$\approx$
$\mathbf{z}_{k+1}$
$\hat{\mathbf{x}}_k$
P_k
Prediction
$\hat{\mathbf{x}}^-_{k+1}$
P^-_{k+1}
$\hat{\mathbf{z}}_{k+1}$
Update
$\hat{\mathbf{x}}^+_{k+1}$
P^+_{k+1}
$\mathbf{x}_{k+1} = \phi(\mathbf{x}_k) + \mathbf{w}_k$
$\mathbf{z}_{k+1} = \mathbf{h}(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1}$

Figure 1.1: Kalman Filter Scheme. Image from [16]

.

1. **Prediction Step**: the equations of motion get linearized around the current mean at time step k and they are used to predict the mean and covariance of the state vector and the measurements at time step $k+1$.

2. **Update Step**: The information of the measurements acquired at time step $k+1$ is given to the filter which uses the residuals between the predicted and the acquired measurements to update the estimate by correcting the predicted state vector to the updated one.

The Kalman filter uses this structure of "prediction-update" to determine the state of the spacecraft at time $k+1$ with the knowledge of the measurements at the same time step. The filter, at first, uses the process model to predict the state of the spacecraft at time $k+1$, having as input an estimate of the spacecraft state. The prediction is performed in terms of mean and covariance:

$$\hat{\mathbf{x}}(k+1|k) = E\big[\mathbf{f}[\mathbf{x}(k), \mathbf{u}(k), k] + \mathbf{v}(k)|\mathbf{Z}^k\big] \tag{1.3}$$

$$\mathbf{P}(k+1|k) = E\big[\{\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|k)\}\{\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|k)\}^T|\mathbf{Z}^k\big] \tag{1.4}$$

where $\hat{\mathbf{x}}(i|j)$ be the estimate of $\mathbf{x}(i)$ using the observation information up to an including time j, and $\mathbf{Z}^j = [\mathbf{z}(1), \ldots, \mathbf{z}(j)]$ describes the measurements. The conditional covariance of the estimate is $\mathbf{P}(i|j)$.

First, the current state and the associated covariance are transformed through the state transition and the observation equations. These values $\hat{\mathbf{x}}(k+1|k)$ and $\mathbf{P}(k+1|k)$ are the *predicted* quantities since they represent the estimation of the state of the system at time $k+1$ given the knowledge of all observations up to time k. In the second step, the *update*, the system is observed at time $k+1$ and the filter uses the observation information (the measurements) to correct the predicted quantities, producing the estimate $\hat{\mathbf{x}}(k+1|k+1)$ and $\mathbf{P}(k+1|k+1)$. The *update* part of the filter algorithm is here presented:

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)n(k+1) \tag{1.5}$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{K}(k+1)\mathbf{P}_{zz}(k+1)\mathbf{K}^T(k+1) \tag{1.6}$$

$$n(k+1) = \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1|k) \tag{1.7}$$

$$\mathbf{K}(k+1) = \mathbf{P}_{xz}(k+1|k)\mathbf{P}_{zz}^{-1}(k+1|k) \tag{1.8}$$

where vector $n(k+1)$ is the innovation, which is evaluated as the difference between the actual observation at time k, $\mathbf{z}(k+1)$, and the predicted mean observation $\hat{\mathbf{z}}(k+1|k)$. Furthermore, $\mathbf{K}(k+1)$ is the Kalman gain matrix, evaluated with the information of the covariance matrix of the state measurements $\mathbf{P}_{zz}(k+1|k)$ and the cross-covariance matrix of the state and the measurement $\mathbf{P}_{xz}(k+1|k)$. It can be seen that the *update* equations are only a function of the predicted values of the first two moments of $\mathbf{x}(k)$ and $\mathbf{z}(k)$.

Generally $\mathbf{f}[\cdot]$ and $\mathbf{h}[\cdot]$ are nonlinear, thus the drawback of the EKF is the accuracy of the linearization in highly-nonlinear systems, which can be related to the dynamical model or the use of low measurement frequencies. Therefore, techniques such as the Unscented Kalman filter (UKF) and high order filters have been developed. While the UKF relies on the UT, the DA-based EKF that will be presented in this thesis relies on the use of DA in the high-order extended Kalman filter initially introduced by Park and Scheeres [32] [33]. This filter is more accurate that the basic EKF since the prediction step relies on the fully nonlinear mapping of the first two moments. Parks and Scheeres implemented a semi-analytical orbit uncertainty propagation technique, that solves the higher-order Taylor series terms that describe the nonlinear motion and use the resulting polynomials to analytically map the initial uncertainties. However, the high order tensors may be difficult to be derived due to the hard and complex computational work required. This problem can be solved with the introduction of DA that, by replacing the classical implementation of real algebra with the implementation of a new algebra of Taylor polynomials, enables the computation of the Taylor expansion of any function f of n variables up to an arbitrarily order. The consequence is a relevant improvement in the computational effort to Taylor expand the solution of an ordinary differential equation (ODE).

It has been stated that the classical sequential state estimation techniques have limits operating with nonlinear uncertainties problems. The introduction of DA and UT, and later on DA inside UT, represents a clever way to overcome the classical EKF limitations. The resulting filters will then assess their performance on the relative pose estimation problem to be studied and compared.

1.1.1 Differential Algebra

Differential algebraic (DA) techniques are here proposed as a valuable tool to implement the DA-based EKF and the DA-based UKFDA, in order to obtain not only a higher-order filter, but also a computationally efficient one. Differential algebra supplies the tools to compute the derivatives of functions within a computer environment [43]. More specifically, by substituting the classical implementation of real algebra with the implementation of a new algebra of Taylor polynomials, any function f of n variables is expanded into its Taylor polynomial up to an arbitrary order m. This has a strong impact when the numerical integration of an ordinary differential equation (ODE) is performed by means of an arbitrary integration scheme. Any integration scheme is based on algebraic operations, involving the evaluation of the ODE right hand side at several integration points. Therefore, starting from the DA representation of the initial conditions and carrying out all the evaluations in the DA framework, the flow of an ODE is obtained at each step as its Taylor expansion in the initial conditions. The accuracy of the Taylor expansion can be kept arbitrarily high by adjusting the expansion order. So, in the DA-based EKF and UKFDA presented in this thesis the propagation of the mean trajectory is carried out in the DA framework.

The key idea of DA is to replace all algebraic operations between numbers by ones that act on a suitably chosen subset of polynomials instead [2]. Thus, algebraic operations, in the space of truncated Taylor polynomials, are defined such that they approximate the operations of the function space $\mathcal{C}^r(0)$ of r times differentiable functions around point 0. In detail, each single operation is defined to result in the truncated Taylor expansion of the correct result computed on the function space $\mathcal{C}^r(0)$. Figure 1.2 shows an example between working within a function in $\mathcal{C}^r(0)$ or in the DA arithmetic.

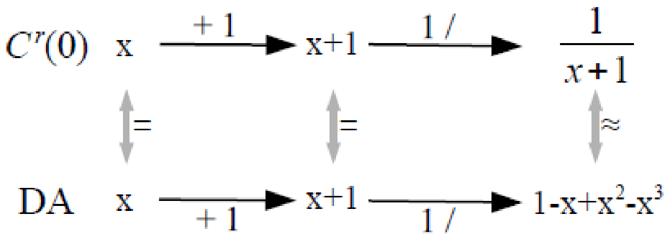

Figure 1.2: Evaluation of the expression 1/(1+x) in $\mathcal{C}^r(0)$ and in the DA arithmetic.

Since the accuracy of the Taylor expansion can be kept arbitrarily high by adjusting the expansion order, the approach of classical pointwise simulations can be enhanced by replacing thousands of integrations with evaluations of the Taylor expansion of the flow. As a result, the computational time reduces considerably without any significant loss in accuracy [3].

1.1.2 Unscented Transformation

The unscented transformation (UT) tries to solves the computational drawbacks of the moments propagation through linearization [46]. The state distribution is again represented as a Gaussian Random Variable (GRV), but it gets specified by using a minimal set of carefully chosen sample points, the so-called sigma points. These sample points completely capture the true mean and covariance of the GRV, and when propagated through the true nonlinear system, they capture the posterior mean and covariance accurately to the 3rd order (Taylor series expansion) for any non-linearity. Non-Gaussian inputs approximation are accurate at least to the second order.

The UT then calculates the statistics of a random variable which undergoes a nonlinear transformation. The covariance is expressed thanks to the set of sigma points which, in addition to the mean, get propagated through the nonlinear function. The mean and covariance of the transformed variable are then approximated using a weighted sample mean and covariance of the posterior sigma points. Figure 1.3 presents a simple 2-dimensional example

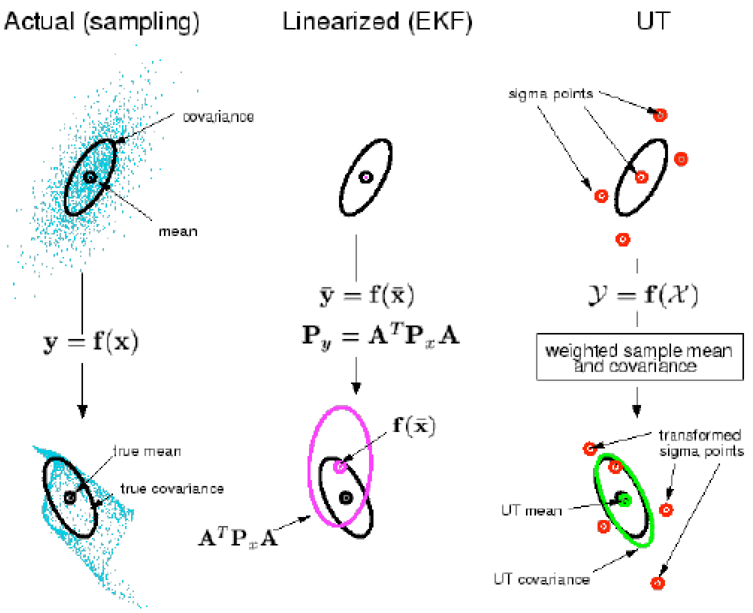
Actual (sampling)
Linearized (EKF)
UT
covariance
mean
$\mathbf{y} = \mathbf{f}(\mathbf{x})$
true mean
true covariance
$\bar{\mathbf{y}} = \mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{P}_y = \mathbf{A}^T\mathbf{P}_x\mathbf{A}$
$\mathbf{f}(\bar{\mathbf{x}})$
$\mathbf{A}^T\mathbf{P}_x\mathbf{A}$
sigma points
$\mathcal{Y} = \mathbf{f}(\mathcal{X})$
weighted sample mean and covariance
transformed sigma points
UT mean
UT covariance

Figure 1.3: Representation of the linearization and UT approaches to moments propagation. Taken from [46].

to underline the superior performance of the UT with respect to linearization. The left plots show the true mean and covariance propagation using a Monte Carlo sampling; the central plots show the results using a linearization approach as in the EKF; the right plots show the performance of the UT. In the figure it can be seen that only 5 sigma points have been used: therefore the UT method differs substantially from the general "sampling" methods (such as particle filters still based on Monte Carlo type sampling) which require orders of magnitude more sampling points in the attempt to propagate an accurate distribution of the state [15].

1.2 Relative Pose Estimation

One of the main space applications for filters is the relative pose estimation during the rendezvous between two separate spacecraft. A particularly challenging application is the on-board autonomous guidance, navigation and control (GNC) when one of the spacecraft is uncooperative [49]. The autonomous rendezvous and docking requires that two spacecraft start at a remote distance, come together into a common orbit, rendezvous, dock, and control the new combined spacecraft in both orbit and attitude. Therefore a sequential state estimation (i.e. a filter) is requested since the spacecraft that is approaching, called the *chaser*, needs to know the relative position and orientation, i.e. the pose, of the spacecraft that has been approached, called the *target*. The chaser is always the active spacecraft, while the target can be completely passive, and its role is just to remain reasonably stable while the chaser does all the work. The target, thus, can be either cooperative or uncooperative: in the first case, it is expected a target active participation in the docking maneuver, while in the second case the spacecraft has neither active or passive equipment to help the approaching process. A typical example of the latter is the case of a chaser approaching the target during an active debris removal (ADR) mission [51].

This thesis applies the implemented filters to the problem of capturing the satellite ENVISAT (see Figure 1.4) with a chaser satellite. The approaching maneuver itself is one of the main challenge in this problem since the chaser must study and correctly predict the attitude of the target which, in general, tends to have a tumbling motion. As a consequence, there is the continue effort to create an effective and accurate autonomous GNC strategy by enhancing its reliability by increasing the accuracy of the estimation process and the robustness of the control laws to noise and system errors. Autonomous GNC technologies for rendezvous and docking demand accurate, real-time measurements and estimations of relative range and attitude [24].

The knowledge of the target features and characteristics, in addiction to the many information that can be found on the web and from the space community, classifies this particular application as a model-based estimation technique. Thanks to the availability of satellites models and CAD descriptions, model-based techniques can take advantage of a priori knowledge of the target whose pose and motion are to be estimated. Regardless the kind of measurements taken, the estimation method shall include filtering due to the presence of noise and errors. Therefore, the relative pose estimation and attitude prediction problem must update, each time, the state and covariance of the DOF that describe the motion and must also update the measurement

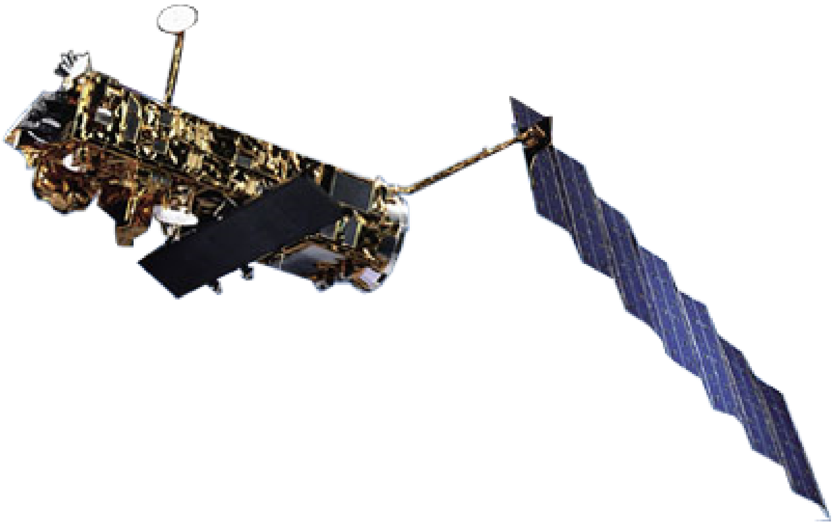

Figure 1.4: ENVISAT

and state vector estimation. The state vector must include both the translational and the rotational motion.

The aim of this work is to assess the performance and on-board applicability of the DA-based Extended Kalman Filter and the Unscented Kalman Filter algorithms with the target application of estimating the relative pose between two spacecraft during a rendezvous manoeuvre. The filters will be proven to achieve excellent accuracy level and their performance will be compared one another in terms of precision, robustness and computational time. At a later stage, the filters must be proven to be robust to failures, at different operational frequencies.

1.3 Thesis Overview

A quick work outline is here presented.

In chapter 2, an introduction to the Differential Algebra is presented, focusing on its theoretical framework and the opportunity of using DA as a new algebra to deal with computer operations.

Chapter 3 compares different propagation techniques of mean and covariance. The DA framework and the Unscented Transformation are explained and presented as a valid improvement with respect to the linearization technique. A transformation from polar to Cartesian coordinates is used as an example to show the improvements of the described techniques. The propagation of uncertainties is the key idea upon the filters are based on. Chapter 3 purpose is to describe and show how DA and UT work with moments and how they represent mean and covariance putting attention on the computational request, on the theory, and on the different results that are achieved. Since the filtering statement is based on the knowledge of the moments, only after having in mind a clear idea on how uncertainties are represented and calculated, it is possible to describe the nonlinear filters algorithms.

Chapter 4 describes the filters. After presenting the filter statement, the DA-based High Order Extended Kalman filter, and the Unscented Kalman filter, are described in details providing their algorithms and pointing out their common aspects and differences. Then, a mixed filter that combines both DA and UT is presented. At the end of the chapter, an orbit determination problem in two body dynamics is used as test case to assess the improvements with respect to the basic extended Kalman filter.

Chapter 5 takes the filters and applies them to a more realistic and complex application: the ENVISAT relative pose estimation. After a detailed presentation of the satellite dynamics, a careful presentation of the measurement model (i.e. the ENVISAT markers acquisition, disposition and visibility) is provided. In the last part of the chapter, the results of a robustness and accuracy (Monte Carlo-based) analysis are reported for three different cases: the default acquisition of all the markers, the performance obtained with only to 3 markers, and lastly the presence of failures in markers acquisition.

Chapter 6, at the end, summarizes the work and draws the appropriate conclusions. Moreover, final considerations are provided and future developments are proposed.

CHAPTER 2

Differential Algebra

This chapter is dedicated to the illustration of differential algebra. The theory of differential algebra has been developed by Martin Berz in the late 80's, and the short summary given in the followings takes advantages of his books *Advances in Imaging and Electron Physics* [8] and *Modern Map Methods in Particle Beam Physics* [28].

2.1 Introduction

Differential algebraic (DA) techniques find their origin in the attempt to solve analytical problems by an algebraic approach. Historically, treatment of functions in numerics has been based on the treatment of numbers, and the classical numerical algorithms are based on the mere evaluation of functions at specific points. DA techniques are based on the observation that it is possible to extract more information on a function rather than its mere values. The basic idea is to bring the treatment of functions and the operations on them to the computer environment in a similar way as the treatment of real numbers. Of particular interest is the possibility to efficiently determine Taylor expansions of the flow of differential equations in terms of initial conditions. Referring to Figure 2.1, consider two real numbers a and b: in order to operate in a computer environment, they get usually transformed in their floating points (FP) representation, $\bar{\mathbf{a}}$ and $\bar{\mathbf{b}}$ respectively. Then, given any operation × in the set of the real numbers, an adjoint operation ⊗ is defined in the set of the FP numbers such that the diagram commutes. Consequently, the same result is obtained if the real numbers a and b get transformed in their FP representation and then operating on them in the set of FP numbers, or if the operation is carried out in the set of real numbers and subsequently transform the achieved result in its FP representation (left part of Figure

$$\begin{tikzcd}
a, b \in R \arrow[r, "\mathcal{T}"] \arrow[d, "\times"'] & \overline{a}, \overline{b} \in FP \arrow[d, "\otimes"] \\
a \times b \arrow[r, "\mathcal{T}"'] & \overline{a} \otimes \overline{b}
\end{tikzcd}
\qquad
\begin{tikzcd}
f, g \arrow[r, "\mathcal{T}"] \arrow[d, "\times"'] & F, G \arrow[d, "\otimes"] \\
f \times g \arrow[r, "\mathcal{T}"'] & F \otimes G
\end{tikzcd}$$

Figure 2.1: Analogy between the floating point representation of real numbers in a computer environment (left figure) and the introduction of the algebra of Taylor polynomials in the differential algebraic framework (right figure). Image from [27], as well [42] [3]

2.1). In an analogous way, suppose that two different, sufficiently regular, functions f and g are given. In the framework of DA, the computer operates on them by using their Taylor series expansions, referred respectively as F and G. Therefore, the transformation of real numbers in their FP representation is now substituted by the extraction of the Taylor expansion of f and g up to a selected m order. For each operation in the function space, an adjoint operation in the space of Taylor polynomials is defined such that the corresponding diagram commutes. Therefore, the same result is achieved by extracting the Taylor expansions of f and g and operating on them in the set of Taylor polynomial, as well as operating on f and g in the original space and then extracting the Taylor expansion of the resulting function.

Differential algebra has the strength of being effectively implementable in a computer environment. In this way, the Taylor series coefficients of a function can be obtained up to an arbitrary and specified order m, along with the relative function evaluation, with a fixed amount of effort. The Taylor coefficients of any order for sums and product of functions, as well as scalar products with reals, can be computed from those of summands and factors; therefore, the set of equivalence classes of functions can be endowed with well-defined operations, leading to the so-called truncated power series algebra (TPSA) [6] [7] .

Similarly to the algorithms for floating point arithmetic, the algorithm for functions followed, including methods to perform composition of functions, to invert them, to solve nonlinear systems explicitly, and to treat common elementary functions [4] [5] .

In addition to these algebraic operations, the DA framework is endowed with differentiation and integration operators, therefore finalizing the definition of the DA structure. In the following, the minimal differential algebra for 1-dimensional functions and their first order expansion is explained in details, and some hints on its extension to functions of v variables and to n-th order are given.

2.2 The Minimal Differential Algebra

The simplest non-trivial differential algebra is here described. Consider all ordered pairs (q_0, q_1), with q_0 and q_1 in the set of real numbers. The addiction, scalar multiplication, and vector multiplication are defined as follows.

$$\begin{aligned}(q_0, q_1) + (r_0, r_1) &= (q_0 + r_0, q_1 + r_1)\\ t \cdot (q_0, q_1) &= (t \cdot q_0, t \cdot q_1)\\ (q_0, q_1) \cdot (r_0, r_1) &= (q_0 \cdot r_0, q_0 \cdot r_1 + q_1 \cdot r_0)\end{aligned} \tag{2.1}$$

The ordered pairs with the arithmetic are called ${}_1D_1$. The first two equations are the familiar vector space structure of $\mathbb{R}^2$, whereas the multiplication is similar to the one in the complex number domain; except here $(0, 1) \cdot (0, 1)$ is equal to $(0, 0)$ rather then $(-1, 0)$. The multiplication of vectors is seen to have $(1, 0)$ as the unity element. Furthermore, the multiplication is associative, commutative, and distributive w.r.t the addition. Together, the three operations defined in (2.1) form an algebra. Furthermore, they do form an extension of real numbers, as $(r, 0) + (s, 0) = (r + s, 0)$ and $(r, 0) \cdot (s, 0) = (r \cdot s, 0)$, so that the reals can be included.

However, ${}_1D_1$ does not present all the characteristics of a field, since (q_0, q_1) has an inverse in ${}_1D_1$ if and only if $q_0 \neq 0$. Thus, if $q_0 \neq 0$, then

$$(q_0, q_1)^{-1} = \left(\frac{1}{q_0}, -\frac{q_1}{q_0^2}\right). \tag{2.2}$$

If q_0 is positive, then $(q_0, q_1) \in {}_1D_1$ has a root defined by

$$\sqrt{(q_0, q_1)} = \left(\sqrt{q_0}, \frac{q_1}{2\sqrt{q_0}}\right) \tag{2.3}$$

just as simple arithmetic shows. One of the most important aspects of this algebra is that it has encoded an order compatible with its algebraic operations. Given two elements (q_0, q_1) and (r_0, r_1) in ${}_1D_1$, it is defined as

$$\begin{aligned}&(q_0, q_1) < (r_0, r_1) \;\; if \;\; q_0 < r_0 \;\; or \;\; (q_0 = r_0 \;\; and \;\; q_1 < r_1)\\ &(q_0, q_1) > (r_0, r_1) \;\; if \;\; (r_0, r_1) < (q_0, q_1)\\ &(q_0, q_1) = (r_0, r_1) \;\; if \;\; q_0 = r_0 \;\; and \;\; q_1 = r_1\end{aligned} \tag{2.4}$$

As for any two elements (q_0, q_1) and (r_0, r_1) in ${}_1D_1$ only one of three relation holds, therefore ${}_1D_1$ is said to be totally ordered. The order is compatible with the addition and multiplication; for all (q_0, q_1), (r_0, r_1), $(s_0, s_1) \in {}_1D_1$, it follows $(q_0, q_1) < (r_0, r_1) \Rightarrow (q_0, q_1) + (s_0, s_1) < (r_0, r_1) + (s_0, s_1)$; and $(s_0, s_1) > (0, 0) = 0 \Rightarrow (q_0, q_1) \cdot (s_0, s_1) < (r_0, r_1) \cdot (s_0, s_1)$.

The number $d = (0, 1)$ has the interesting property of being positive but smaller than any positive real number, indeed

$$(0, 0) < (0, 1) < (r, 0) = r \tag{2.5}$$

Due to this reason, d is called an infinitesimal or a differential. In fact, d is such a small number that its square vanishes in ${}_1D_1$. Since for any $(q_0, q_1) \in {}_1D_1$

$$(q_0, q_1) = (q_0, 0) + (0, q_1) = q_0 + d\dot{q}_1 \tag{2.6}$$

the first component is called the real part while the second component takes the name of differential part.

The algebra developed in ${}_1D_1$ becomes a differential algebra with the introduction of a map ∂ from ${}_1D_1$ to itself, and it can be proven that the map is a derivation. Define $\partial : {}_1D_1 \to {}_1D_1$ by

$$\partial(q_0, q_1) = (0, q_1) \tag{2.7}$$

Note that

$$\begin{aligned} \partial\{(q_0, q_1) + (r_0, r_1)\} &= \partial(q_0 + r_0, q_1 + r_1) = (0, q_1 + r_1) \\ &= (0, q_1) + (0, r_1) = \partial(q_0, q_1) + \partial(r_0, r_1) \end{aligned} \tag{2.8}$$

and

$$\begin{aligned} \partial\{(q_0, q_1) \cdot (r_0, r_1)\} &= \partial(q_0 \cdot r_0, q_0 \cdot r_1 + r_0 \cdot q_1) = (0, q_0 \cdot r_1 + r_0 \cdot q_1) \\ &= (0, q_1) \cdot (r_0, r_1) + (0, r_1) \cdot (q_0, q_1) \\ &= \{\partial(q_0, q_1)\} \cdot (r_0, r_1) + (q_0, q_1) \cdot \{\partial(r_0, r_1)\} \end{aligned} \tag{2.9}$$

These two identities hold for every (q_0, q_1), $(r_o, r_1) \in {}_1D_1$. Therefore ∂ is a derivation and $({}_1D_1, \partial)$ is a differential algebra.

It is important to point out what is the most relevant aspect of ${}_1D_1$: it allows the automatic computation of the derivatives. In order to show this fundamental feature, assume two functions f and g; put their values and their derivatives at the origin in the form of $(f(0), f'(0))$ and $(g(0), g'(0))$ as two vectors in ${}_1D_1$; then consider the product

$$(f(0), f'(0)) \cdot (g(0), g'(0)) = (f(0) \cdot g(0), f(0) \cdot g'(0) + f'(0) \cdot g(0)) \tag{2.10}$$

As can be seen, if the derivative of the product $f \cdot g$ is of interest, it is sufficient just to look at the second component of the resulting pair in Equation (2.10); whereas the first component gives the value of the product of the two functions. Therefore, if two vectors contain the values and the derivatives of two functions, their product contains the values and the derivatives of the product function.

Defining the operator [ ] from the space of differential functions to ${}_1D_1$ via

$$[f] = (f(0), f'(0)) \tag{2.11}$$

it holds

$$\begin{aligned} [f+g] &= [f] + [g] && (2.12)\\ [f \cdot g] &= [f] \cdot [g] && (2.13)\\ [1/g] &= [1]/[g] = 1/[g] && (2.14) \end{aligned}$$

by using Equation (2.2). This observation leads to the ability to compute derivatives of different kinds of functions algebraically just by merely applying arithmetic rules in ${}_1D_1$, beginning from the value and the derivative of the identity function.

The following example is now offered: consider the function

$$f(x) = \frac{1}{x + \frac{1}{x}} \tag{2.15}$$

and its derivative

$$f'(x) = \frac{(1/x^2) - 1}{(x + (1/x))^2} \tag{2.16}$$

The function values and the relative derivatives at point $x = 3$ are

$$f(3) = \frac{3}{10}, \quad f'(3) = -\frac{2}{25} \tag{2.17}$$

Evaluating function (2.15) at the ordered pair corresponding to the identity function, i.e. $[x] = (x, 1)$, at point 3, i.e. $(3, 1) = 3 + d$, yields

$$\begin{aligned} f((3,1)) &= \frac{1}{(3,1) + 1/(3,1)} = \frac{1}{(3,1) + (1/3, -1/9)}\\ &= \frac{1}{(10/3, 8/9)} = \left(\frac{3}{10}, -\frac{8}{9}/\frac{100}{9}\right) = \left(\frac{3}{10}, -\frac{2}{25}\right) \end{aligned} \tag{2.18}$$

As the equation shows, after the function evaluation, the real part of the result is the value of the function at $x = 3$, whereas the differential part is the value of the derivative of the function at $x = 3$. This is simply justified by applying the relations (2.12),(2.13) and (2.14).

$$\begin{aligned}[f(x)] &= \left[\frac{1}{x+1/x}\right] = \frac{1}{[x+1/x]} \\ &= \frac{1}{[x]+[1/x]} = \frac{1}{[x]+1/[x]} \\ &= f([x]) \end{aligned} \tag{2.19}$$

Since, for a real number x, $[x] = (x, 1) = x + d$, and $[f(x)] = (f(x), f'(x))$, apparently

$$(f(3), f'(3)) = f((3+d)) \tag{2.20}$$

The method can be generalized to allow the treatment of common intrinsic functions, like trigonometric or exponential functions, by setting

$$\begin{aligned} g_i([f]) &= [g_i(f)] \qquad or \\ g_i((q_0, q_1)) &= (g_i(q_0), q_1 g_i'(q_0)) \end{aligned} \tag{2.21}$$

Therefore, by virtue of Equations (2.1) and Equation (2.21) any function f representable by finitely many additions, subtractions, multiplications, divisions, and intrinsic functions in ${}_1D_1$ satisfies the important relationship

$$[f(x)] = f([x]) \tag{2.22}$$

Moreover, note that $f(r+d) = f(r) + d \cdot f'(r)$ resemble $f(x+\Delta x) \approx f(x) + \Delta x \cdot f'(x)$, in which the approximation becomes increasingly more refined for smaller Δx.

2.3 The Differential Algebra ${}_nD_v$

The algebra described in this section is introduced to compute the derivatives up to an order n of a function with v variables. Similarity as before, it is based on considering the space $\mathcal{C}^n(\mathbb{R}^v)$, i.e. the collection of n times continuously differentiable functions on $\mathbb{R}^v$. For this space an equivalent relation is introduced. For any f and $g \in \mathcal{C}^n(\mathbb{R}^v)$, $f =_n g$ if and only if $f(0) = g(0)$ and all the partial derivatives of f and g agree at 0 up to order n. The relation $=_n$ satisfies

$$\begin{aligned} f &=_n f \quad \forall \;\; f \in \mathcal{C}^n(\mathbb{R}^v) \\ f =_n g \Rightarrow g &=_n f \quad \forall \;\; f, g \in \mathcal{C}^n(\mathbb{R}^v) \\ f =_n g \; and \; g =_n h \Rightarrow f &=_n h \quad \forall \;\; f, g, h \in \mathcal{C}^n(\mathbb{R}^v) \end{aligned} \tag{2.23}$$

Thus, $=_n$ is an equivalence relation and a new set can be created: the equivalence class $[f]$ of the function f, which groups together all the elements that are related to f. The resulting equivalence classes are often referred to as DA vectors or DA numbers. Intuitively, each of these classes is then specified by a particular collection of partial derivatives all in v variables up to order n. This class is called ${}_nD_v$.

If the values and the derivatives of the two functions f and g are known, the corresponding values and derivatives of $f + g$ and $f \cdot g$ can be inferred. Therefore, the arithmetic on the classes in ${}_nD_v$ can be introduced via

$$[f] + [g] = [f + g] \tag{2.24}$$

$$t \cdot [f] = [t \cdot f] \tag{2.25}$$

$$[f] \cdot [g] = [f \cdot g] \tag{2.26}$$

Under this operations, ${}_nD_v$ becomes an algebra.

For each $k \in 1, \ldots, v$ define the map ∂_k from ${}_nD_v$ to ${}_nD_v$ for f via

$$\partial_k[f] = \left[p_k \cdot \frac{\partial f}{\partial x_k}\right] \tag{2.27}$$

where

$$p_k(x_1, \ldots, x_k) = x_k \tag{2.28}$$

projects out the k-th component of the identity function. It is easy to show that, for all $k = 1, \ldots, v$ and for all $[f]$, $[g] \in {}_nD_v$,

$$\partial_k([f] + [g]) = \partial_k[f] + \partial_k[g] \tag{2.29}$$

$$\partial_k([f] \cdot [g]) = [f] \cdot (\partial_k[g]) + [g] \cdot (\partial_k[f]) \tag{2.30}$$

Therefore, ∂_k is a derivation for all k, and hence $({}_nD_v, \partial_1, \ldots, \partial_k)$ is a differential algebra.

The dimension of ${}_nD_v$ is now assessed. Define the special numbers d_k as follows:

$$d_k = [x_k] \tag{2.31}$$

Observe that f lies in the same class as its Taylor polynomial $\mathcal{T}_f$ of order n around the origin; they have the same function values and derivatives up to order n. Therefore,

$$[f] = [\mathcal{T}_f] \tag{2.32}$$

Denoting the Taylor coefficients of the Taylor polynomial $\mathcal{T}_f$ of f as $c_{j_1,\ldots,j_v}$, it follows

$$\mathcal{T}_f(x_1, \ldots, x_v) = \sum_{j_1+\cdots+j_v \leq n} c_{j_1,\ldots,j_v} \cdot x_1^{j_1} \cdots x_v^{j_v} \tag{2.33}$$

with

$$c_{j_1,\ldots,j_v} = \frac{1}{j_1! \cdots j_v!} \cdot \frac{\partial^{j_1+\cdots+j_v} f}{\partial x_1^{j_1} \cdots \partial x_v^{j_v}} \tag{2.34}$$

and thus

$$\begin{aligned}[f] = [\mathcal{T}_f] &= \left[\sum_{j_1+\cdots+j_v \leq n} c_{j_1,\ldots,j_v} \cdot x_1^{j_1} \cdots x_v^{j_v} \right] \\ &= \sum_{j_1+\cdots+j_v \leq n} c_{j_1,\ldots,j_v} \cdot d_1^{j_1} \cdots d_v^{j_v} \end{aligned} \tag{2.35}$$

where, in the last step, the properties of Equations (2.24) and (2.26) have been used. Therefore, the set $\{1, d_k : k = 1, 2, \ldots, v\}$ generates ${}_nD_v$, as any element of ${}_nD_v$ can be obtained from 1 and the d_k via addition and multiplication. Therefore, as an algebra, ${}_nD_v$ has $(v+1)$ generators, and the terms $d_1^{j_1} \cdots d_v^{j_v}$ form a basis for the vector space ${}_nD_v$. It is shown in [28] that the number of basic elements is defined in the following equation, which defines the dimension of ${}_nD_v$.

$$\mathbf{N} = \frac{(n+v)!}{n!v!} \tag{2.36}$$

Similarly to the structure ${}_1D_v$, ${}_nD_v$ can be ordered, and the d_k, being smaller than any other real number, are infinitely small or infinitesimal. Furthermore, a fixed point theorem for contracting operations in ${}_nD_v$ exists, which enables the evaluation of the square roots, the quotient, and the inversion of Taylor polynomials through iterative process based on a finite number of steps [28]. Once the function composition and the elementary functions, like trigonometric or exponential functions, are introduced in ${}_nD_v$, the derivatives of any function f belonging to $\mathcal{C}^n(\mathbb{R}^v)$ can be computed up to order n in a fixed amount of effort by applying

$$[f(x_1, \ldots, x_v)] = f([x_1, \ldots, x_v]) = f(x_1 + d_1, \ldots, x_v + d_v) \tag{2.37}$$

The differential algebra sketched in this section was implemented by M. Berz and K. Makino in the software COSY INFINITY [30]. The filter presented in this thesis have been implemented using the "Differential Algebra Computational Engine", DACE software. The DACE module is composed by three different sub-modules: the DACE Fortran core, the C++ interface, and the Matlab interface (see [35]).

CHAPTER 3

Moments Propagation Techniques

The covariance is the main parameter that is adopted to describe how the distribution of points is spread around their mean value [38] [44] [40]. The analysis of the covariance ellipse gives information about the system time behaviour and also about how the system state responds to a transformation. By describing the distribution of points by only the mean and covariance, it is assumed a Gaussian distribution, characteristic that is no more valid after a nonlinear transformation.

Within any filtering process, the state of the system undergoes a transformation, and this propagation is often nonlinear. The transformation of the covariance is then one of the most critical filtering steps: given a set of random points whose distribution is accurately described by their mean and covariance, the distribution resulting from their mapping through the nonlinear transformation cannot be exactly described by providing only its mean and covariance.

This chapter describes how different techniques try to propagate the covariance under nonlinear transformations. The Monte Carlo method (MC) is used as benchmark for the assessment of the performance of the linearization, the propagation with DA, and the Unscented Transformation (UT).

3.1 Uncertainty Propagation

Suppose $\mathbf{x}$ is a random variable with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$. A second variable $\mathbf{y}$ is related to $\mathbf{x}$ through the nonlinear function $\mathbf{f}$, i.e.

$$\mathbf{y} = \mathbf{f}[\mathbf{x}] \tag{3.1}$$

The aim is to see how different methods such as MC, Linearization, DA and UT calculate the mean $\bar{\mathbf{y}}$ and the covariance $\mathbf{P}_{yy}$.

The nature of $\mathbf{f}[\cdot]$ is the driver to calculate the statistics of $\mathbf{y}$ by determining the density function of the transformed distribution and evaluating the statistics from that distribution. In the case of a linear function, the exact close solution can be derived. However, such solutions may be inaccurate since many applications have to deal with high non-linearities in the transformation function. Therefore some approximate methods must be used: these methods should be *efficient* and *unbiased* and they should yield *consistent* statistics. A performance analysis is mandatory before the description of how different filters work in order to have a clear idea of the covariance propagation concepts these methods are based on.

The transformed statistics are consistent if the equality

$$\mathbf{P}_{yy} - E[\{\mathbf{y} - \bar{\mathbf{y}}\}\{\mathbf{y} - \bar{\mathbf{y}}\}^T] \geqslant 0 \tag{3.2}$$

holds, where $E[\cdot]$ represents the expected value. If the statics are not consistent, then the value of the transformed covariance is under-estimated; hence, this is such an important propriety for the validity of the transformation method. A Kalman filter that uses an inconsistent set of statistics will give too much weight on the information and, as a consequence, it will under estimate the covariance, leading the filter to the strong possibility of diverging. Ensuring that the transformation is consistent guarantees that the filter will be consistent as well. However, consistency does not necessarily imply accuracy since the calculated value of $\mathbf{P}_{yy}$ might greatly exceed the actual error. Therefore, it is better, and desirable, to work with an efficient transformation that minimises the left hand side of Equation (3.2). Moreover, it is desirable that the estimate is *unbiased.*

A common test case is adopted in the following sections to compare the results of different propagation techniques. The selected example is the same used by Julier and Uhlmann in [22] [23] since it is both simple and generates a highly nonlinear density function of the transformed points distribution.

A mobile robot detects beacons in its environment using a range-optimised sonar sensor. The sensor returns the polar information, thus the range r and the bearing θ, and these information are sent to be converted to Cartesian coordinates. Therefore the transformation is:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r\cos\theta \\ r\sin\theta \end{pmatrix}$$

The real location of the target is $(0, 1)$. This sonar has a fairly good range accuracy, with a standard deviation of 2 cm, which is traded off to give a very poor bearing measurement, with a standard deviation of 15°. The difficulty of the transformation is here enhanced because the sonar physical proprieties, especially the large bearing uncertainty, do not make possible the assumption of local linearity.

3.2 Monte Carlo

Monte Carlo methods are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. Their essential idea is using randomness to solve problems that might be deterministic in principle. The general idea is to use a statistical analysis on an high number of samples to study how the system responds to uncertainties on its variables. The drawback of MC methods is that they require strong computational work and thus they are not so efficient in problems such as filtering where time is one of the key drivers. However, MC is here used as benchmark for the problem of uncertainty propagation: as it is able to provide accurate values for the mean and covariance if a proper number of samples is used.

Referring to the sonar example, MC method generates a random distribution of points once the mean and the standard deviations of $\mathbf{x}$ are given. Then, each single point is transformed according to the function rules and the mean and the covariance are evaluated on the resulting distribution of samples. As expected, this method can work in an efficient way only for simple functions since the propagation of each sample would require a significant computational time in the propagation of dynamical systems, as required in the filtering process. The MC method has been applied to the sonar problem and Figure 3.1 shows the transformed sampling points distribution (a total of 1e6 samples have been used) and the corresponding location of the mean with the relative covariance ellipse at 1σ. The 1σ contour is the locus of point where the Equation $\mathbf{y} : \{(\mathbf{y} - \bar{\mathbf{y}})\mathbf{P}_{\mathbf{y}}^{-\mathbf{1}}(\mathbf{y} - \bar{\mathbf{y}}) = \mathbf{1}\}$ is satisfied and it is a good graphical representation of the size and orientation of $\mathbf{P}_{yy}$. The location of the points shows how this specific example perfectly suits the problem of the impossibility to apply the local linearity assumption since the distribution is highly asymmetric.

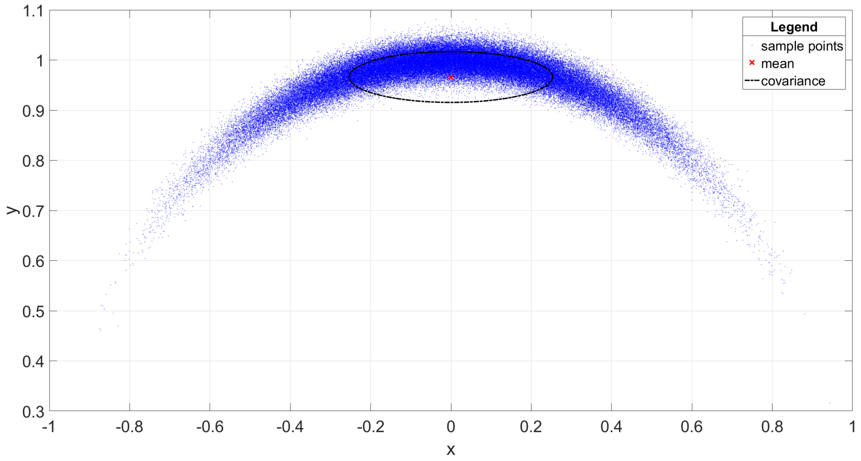

Figure 3.1: Monte Carlo Method 1e6 points: mean and 1σ covariance

3.3 Linearization

The first, and most intuitive solution that comes in mind when dealing with uncertainty propagation is to perform a local linearization in the area of the points to be propagated in order to try to reduce the computational burden of the propagation. Usually, the mean $\bar{\mathbf{x}}$ is used as a reference point for the Taylor expansion. The problem of developing a consistent, efficient and unbiased transformation procedure can be seen by looking at the Taylor expansion of Equation (3.1) about $\bar{\mathbf{x}}$. The series is expressed by Equation (3.3) where $\delta\mathbf{x}$ is a zero mean Gaussian variable with covariance $\mathbf{P}_{xx}$.

$$\begin{aligned} \mathbf{f} &= \mathbf{f}[\bar{\mathbf{x}} + \delta\mathbf{x}] \\ &= \mathbf{f}[\bar{\mathbf{x}}] + \mathbf{\frac{1}{2}\nabla^2 f}\Big|_{\bar{\mathbf{x}}}\delta\mathbf{x^2} + \mathbf{\frac{1}{3!}\nabla^3 f}\Big|_{\bar{\mathbf{x}}}\delta\mathbf{x^3} + \mathbf{\nabla^4 f}\Big|_{\bar{\mathbf{x}}}\delta\mathbf{x^4} + ... \end{aligned} \tag{3.3}$$

Each $\mathbf{\nabla^n f}\delta\mathbf{x^n}$ term represents the appropriate nth order term in the multi-dimensional Taylor Series. At this point, variable $\mathbf{y}$ can be calculated and expressed in terms of mean and covariance by evaluating expectation of the Taylor series. It can be shown that:

$$\bar{\mathbf{y}} = \mathbf{f}[\bar{\mathbf{x}}] + \frac{1}{2}\nabla^2\mathbf{f}\Big|_{\bar{\mathbf{x}}}\mathbf{P}_{xx} + \frac{1}{2}\nabla^4\mathbf{f}\Big|_{\bar{\mathbf{x}}}E[\delta\mathbf{x}^4] + ... \tag{3.4}$$

$$\begin{aligned} \mathbf{P}_{yy} = \nabla\mathbf{f}\Big|_{\bar{\mathbf{x}}}\mathbf{P}_{yy}(\nabla\mathbf{f}\Big|_{\bar{\mathbf{x}}})^T + \frac{1}{2\cdot 4!}\nabla^2\mathbf{f}\Big|_{\bar{\mathbf{x}}}(E[\delta\mathbf{x}^4] - E[\delta\mathbf{x}^2\mathbf{P}_{yy}] - E[\mathbf{P}_{yy}\delta\mathbf{x}^2]+ \\ +\mathbf{P}_{yy}^2)(\nabla^2\mathbf{f}\Big|_{\bar{\mathbf{x}}})^T + \frac{1}{3!}\nabla^3\mathbf{f}\Big|_{\bar{\mathbf{x}}}E[\delta\mathbf{x}^4](\nabla\mathbf{f}\Big|_{\bar{\mathbf{x}}})^T + ... \end{aligned} \tag{3.5}$$

Equation (3.3) shows how the nth order term in the series for $\bar{\mathbf{x}}$ is a function of the nth order moments of $\mathbf{x}$ multiplied by the nth order derivatives of $\mathbf{f}[\cdot]$ evaluated at $\mathbf{x} = \bar{\mathbf{x}}$. If the moments and derivatives can be evaluated correctly up to the nth order, then the mean is correct up to the nth order as well. This is the power of the DA, which can compute the Taylor series up to an arbitrary order and then tune the accuracy of the resulting polynomial. The same reasoning works for the covariance as well.

If the Taylor series converges, each term has an impact on the computed moments that decreases with the order: the lowest order terms in the series are likely to have the greatest influence since higher order terms tend to have smaller coefficients as the series order increases.

The linearization technique assumes that the second order terms of $\delta\mathbf{x}$ in Equation (3.3) can be neglected. Under this assumption, the relations that connect the mean and the covariance of variable $\mathbf{x}$ (before the transformation) to variable $\mathbf{y}$ (after the transformation) are expressed by Equation (3.6) and Equation (3.7).

$$\bar{\mathbf{y}} = \mathbf{f}[\bar{\mathbf{x}}] \tag{3.6}$$

$$\mathbf{P}_{yy} = \nabla\mathbf{f}\mathbf{P}_{xx}(\nabla\mathbf{f})^T \tag{3.7}$$

By comparing Equations (3.6) and (3.7) with Equations (3.4) and (3.5) it is clear that the linearization method, with its approximations, is accurate only if the second and higher orders terms for the mean, and the fourth and higher order terms for the covariance, are negligible.

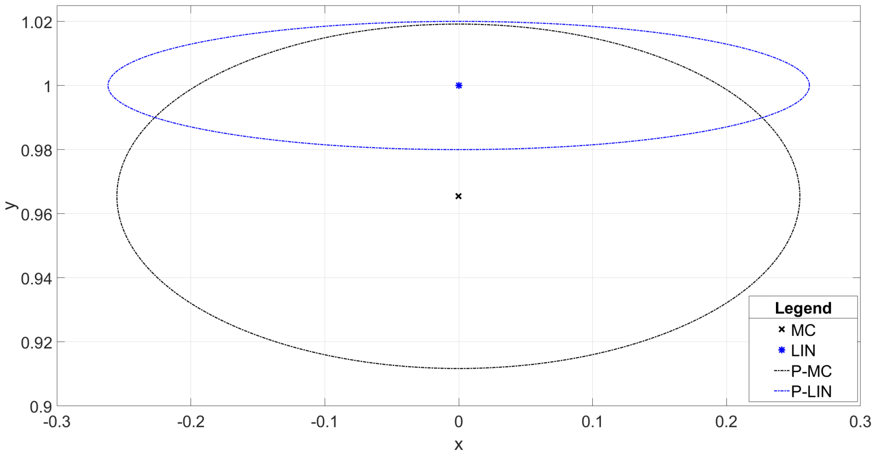

Figure 3.2: Mean and Covariance propagation with Linearization Method

As can be seen in Figure 3.2, the linearized transformation is not capable of accurately describing the covariance of the sample points distribution after the transformation. The figure shows a result inconsistent and biased: this bias is mostly in the range direction, since the linearized mean estimates a position at $y = 1$ whereas in reality it is at $y = 0.967$. The bias error, which arises from the transformation process itself, means that the same error, with the same sign, will be committed each time a coordinate transformation takes place. Moreover, the error associated the linearization is over 1,5 times the standard deviation of the range measurement. Looking at the shape of the 1σ line, it can be seen that the transformation is inconsistent even if there would be no bias. The covariance ellipse is not long enough in the range direction: therefore, the estimate error is not only localized in the mean position, but it influences also the mean squared error, which is much smaller than the true value.

The Kalman Filter (KF) can be applied to nonlinear systems only if a consistent set of predicted quantities can be calculated. These quantities are derived by projecting the prior estimate through the dynamical system whose state is to be estimated. Thus, the linearization, as applied in the EKF, can be inadequate for highly nonlinear systems, however, the alternative approaches tend to show higher and often unaffordable computational complexity Therefore, there is a strong need for a method that is provably more accurate than linearization but, at the same time, does not increase drastically the computational burden. The EKFDA and UKFDA have been developed to meet these needs.

3.4 Differential Algebra

Differential algebraic (DA) techniques [16] [43] are a valuable tool to obtain a higher-order, computationally efficient filter. By enabling the computation of Taylor expansions of arbitrary order, DA is able to operate directly on Equations (3.4) and (3.5), without introducing the limitation of truncating the expansion to the first order. In fact, the linear propagation described in the previous section can be seen as a special case of the more general DA approach, i.e. the one corresponding to order $m = 1$. The accuracy of the Taylor expansion can be kept arbitrarily high by adjusting the expansion order.

Thanks to its properties, DA can be exploited to compute the moments of the transformed probability density function (e.g., mean, covariance, skewness, and kurtosis) by applying the expectation operation to the polynomial maps [43] [16]. In the restricted case of variables with Gaussian random distributions, the moments can be completely described by the first two moments (i.e. mean and covariance) and can be easily computed in terms of the covariance matrix using Isserlis formula on the monomials of the Taylor polynomial [18].

Considering the random variable $\mathbf{x}$ with a probability density function $p(\mathbf{x})$ and the second variable $\mathbf{y}$ related to $\mathbf{x}$ according Equation (3.1), the problem is to calculate a consistent estimate of the main cumulants of the transformed probability density function $p(\mathbf{y})$. As said before, since $\mathbf{f}$ is a generic nonlinear function, this formulation includes a wide range of problems involving uncertainty propagation.

Define independent variable $\mathbf{x}$ as DA variable

$$[\mathbf{x}] = \bar{\mathbf{x}} + \delta\mathbf{x} \tag{3.8}$$

where $\bar{\mathbf{x}}$ is the initial mean, and implement the non linear transformation in ${}_kD_n$ (see Chapter 2). The result is the Taylor expansion of the initial solution with respect to deviation $\delta\mathbf{x}$ of the independent variable

$$[\mathbf{y}] = \mathbf{f}([\mathbf{x}]) = \bar{\mathbf{y}} + \mathcal{M}_y(\delta\mathbf{x}) = \sum_{p_1+\cdots+p_n\leq k} \mathbf{c}_{p_1\ldots p_n} \cdot \delta x_1^{p_1} \cdots \delta x_n^{p_n} \tag{3.9}$$

where $\bar{\mathbf{y}}$ is the zeroth order term of the expansion map, and $\mathbf{c}_{p_1\ldots p_n}$ are the Taylor coefficients of the resulting Taylor polynomial

$$\mathbf{c}_{p_1\ldots p_n} = \frac{1}{p_1!\cdots p_n!} \cdot \frac{\partial^{p_1+\cdots+p_n}}{\partial x_1^{p_1}\cdots\partial x_n^{p_n}} \tag{3.10}$$

Note that the case $k = 1$ corresponds to the ordinary first order statistic propagation (i.e. the approximation corresponding to the linearized model) where $\mathbf{c}_{p_1\ldots p_n}$ are elements of the well-known state transition matrix. The evaluation of Equation (3.9) for a selected value of $\delta\mathbf{x}$ supplies the k^{th} order Taylor approximation of the solution $\mathbf{y}$ corresponding to the displaced independent variable. Of course, the accuracy of the expansion map is function of the expansion order and can be controlled by tuning it. Working in ${}_kD_n$ allows the computation of derivatives up to order k of functions in n variables and the approximation of the original function in the space of Taylor polynomials. Thus, contrary of other methods such the UT (that are founded on the assumption that it is easier to approximate a probability density function that it is to approximate an arbitrary nonlinear function) the starting point and the foundation of DA is the straightforward approximation of the function itself.

Equation (3.9) is used to efficiently compute the propagated statistics by introducing the Taylor polynomial in the definition of mean value μ and covariance $\mathbf{P}$;

$$\mu_i = E[\mathbf{y}_i] = \sum_{p_1+\cdots+p_n \leq k} \mathbf{c}_{i,p_1\ldots p_n} E\left[\delta x_1^{p_1} \cdots \delta x_n^{p_n}\right] \tag{3.11}$$

$$\mathbf{P}_{ij} = E[(\mathbf{y}_i - \mu_i)(\mathbf{y}_j - \mu_j)] = \sum_{\substack{p_1+\cdots+p_n \leq k \\ q_1+\cdots+q_n \leq k}} \mathbf{c}_{i,p_1\ldots p_n} \mathbf{c}_{j,q_1\ldots q_n} E\left[\delta x_1^{p_1+q_1} \cdots \delta x_n^{p_n+q_n}\right] \tag{3.12}$$

where $\mathbf{c}_{i,p_1\ldots p_n}$ are the Taylor coefficients of the Taylor polynomial describing the i^{th} component of $\mathbf{y}$. Note that in the covariance matrix formula the coefficients $\mathbf{c}_{i,p_1\ldots p_n}$ and $\mathbf{c}_{j,q_1\ldots q_n}$ already include the subtraction of the means terms.

Figure 3.3 shows the DA technique applied to the sonar example. After having initialized variable $\mathbf{x}$, i.e. range and bearing, as DA variables, the variable is now written into its Taylor expansion to the arbitrary selected order. The transformation (Equation (3.1)) is performed and the moments are evaluated. Figure 3.3 is the representation of how DA is able to calculate the first two moments, i.e. mean and covariance, with second order accuracy. The accuracy of the mean location is further improved respect the simple linearization case. Moreover, the covariance ellipse at 1σ is now able

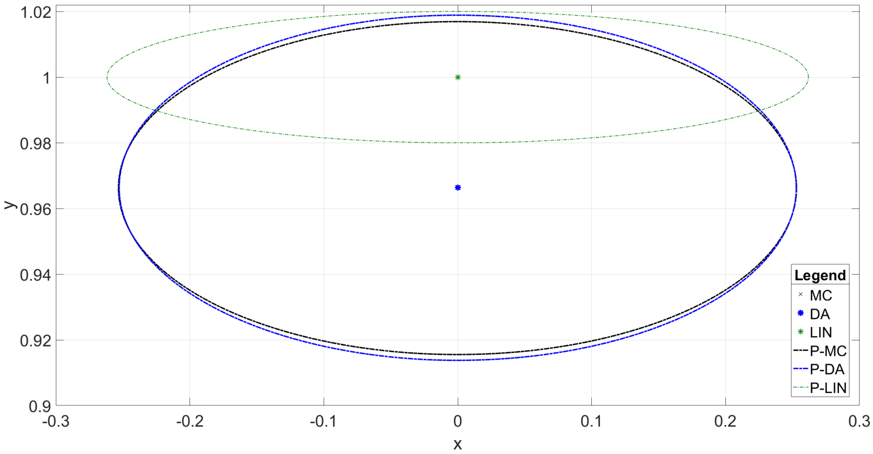

Figure 3.3: Comparison of the mean and covariance computed with the linearization approach and the application of DA

to represent the correct value even in the range direction, where the linearization method was ineffective. The solution presented by DA is coherent and consistent, without the presence of any bias error.

3.5 The Unscented Transformation

The Unscented transformation (UT) is a method to propagate the statistics of a random variable under a nonlinear transformation [46] [22] [45] [21] [23] [20] [41] [39]. The UT is founded on the intuition that *it is easier to approximate a Gaussian (probability) distribution than to approximate an arbitrary nonlinear function or transformation* (see [22], [21] and [23]). Following this statement, looking at Figure 3.4, the aim of the UT is to find a parametrization which accurately describes the mean and the covariance information of the initial variable $\mathbf{x}$ and that, at the same time, permits the direct propagation of the information through the set of nonlinear equations (e.g. functions, transformations,..). This aim can be achieved by generating a discrete distribution of points, the so-called sigma points, so that they have the same first and second (hopefully even higher [34]) moments matched to $\bar{\mathbf{x}}$ and $\mathbf{P}_{xx}$. The nonlinear function (3.1) is then applied to each sigma point. The mean and the covariance of the transformed points distribution $\bar{\mathbf{y}}$ and $\mathbf{P}_{yy}$ are then computed as the estimate of the sigma points distribution after the propagation. At a first sight, this method might resemble the Monte Carlo one (or even a particle filter) but there is an extremely important and fundamental difference. The sigma points are not drawn at random location, but their position is ruled according to a specific and deterministic algorithm based on

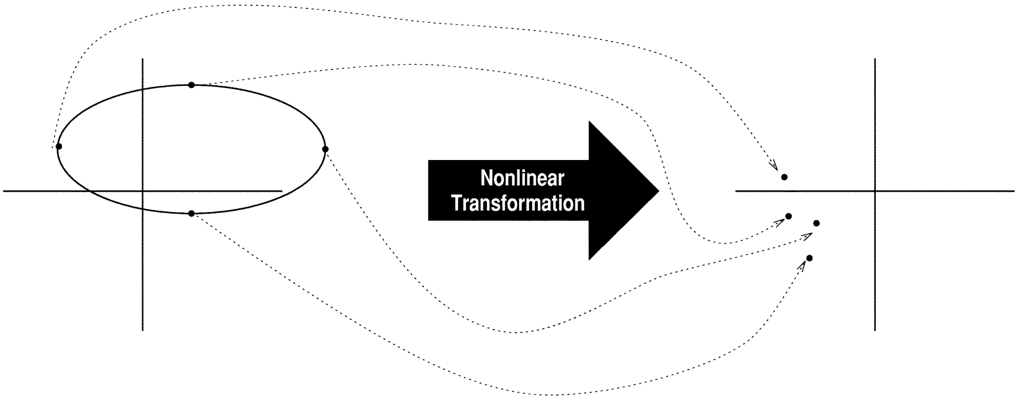

Figure 3.4: Representation of the Unscented Transformation (UT)

the value of the moments. As a result,the non linearity of the distribution is captured using only a fixed, small number of points. Moreover, the sigma points are weighted through parameters in a way that makes them inconsistent with the distribution interpretation of sample points in a particle filter. The weight of a sigma point does not have to lie in the range $[0, 1]$ as for the sample points of the particle filters.

The n-dimensional variable $\mathbf{x}$ with mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$ is approximated by a set of $2n+1$ sigma points. This is the minimal number of points capable of encoding this information. The creation of a bigger set of sigma points is often used in filtering due to the consequent increase of robustness in the Unscented Kalman Filter [45]. A random sampling of points from the distribution, on the other hand, will generally introduce spurious modes in the transformed distribution even if the set of sample points has the correct mean and covariance. In a filtering application these modes will take the form of high frequency noise that may completely obscure the signal.

Being $\mathcal{X}$ the $2n+1$ long sigma points vector of the n-dimensional $\mathbf{x}$ variable, and w the correspondent weight vector; their selection must be such that the following Equations are respected:

$$\sum_{i=0}^{2n} w^{[i]} = 1 \tag{3.13}$$

$$\bar{\mathbf{x}} = \sum_{i=0}^{2n} w^{[i]} \mathcal{X}^{[i]} \tag{3.14}$$

$$\mathbf{P}_{xx} = \sum_{i=0}^{2n} w^{[i]} (\mathcal{X}^{[i]} - \bar{\mathbf{x}})(\mathcal{X}^{[i]} - \bar{\mathbf{x}})^T \tag{3.15}$$

Therefore, there is no unique solution for the sigma points vector and the weights vector. One possibility is to chose the first sigma point as the mean and the remaining ones as indicated in Equations (3.17) and (3.18).

$$\mathcal{X}^{[0]} = \bar{\mathbf{x}} \tag{3.16}$$

$$\mathcal{X}^{[i]} = \bar{\mathbf{x}} + (\sqrt{(n+\lambda)\mathbf{P}_{xx}}\ )_i \qquad i = 1, ..., n \tag{3.17}$$

$$\mathcal{X}^{[i]} = \bar{\mathbf{x}} - (\sqrt{(n+\lambda)\mathbf{P}_{xx}}\ )_{i-n} \qquad i = 1, ..., n \tag{3.18}$$

In Equations (3.18) and (3.17), λ is a scaling parameter that will be later explained, while $(\sqrt{(n+\lambda)\mathbf{P}_{xx}}\ )_i$ is the ith row or column of the matrix square root of $(n+\lambda)\mathbf{P}_{xx}$. Defining $\mathbf{S}$ as $\mathbf{P}_{xx} = \mathbf{SS}$, the matrix square root is computed through diagonalization.

$$\begin{aligned}\mathbf{P}_{xx} &= \mathbf{VDV}^{-1}\\ &= \mathbf{V}\begin{pmatrix} d_{11} & \dots & 0\\ 0 & \ddots & 0\\ 0 & \dots & d_{nn}\end{pmatrix}\mathbf{V}^{-}1\\ &= \mathbf{V}\begin{pmatrix} \sqrt{d_{11}} & \dots & 0\\ 0 & \ddots & 0\\ 0 & \dots & \sqrt{d_{nn}}\end{pmatrix}\begin{pmatrix} \sqrt{d_{11}} & \dots & 0\\ 0 & \ddots & 0\\ 0 & \dots & \sqrt{d_{nn}}\end{pmatrix}\mathbf{V}^{-}1\end{aligned} \tag{3.19}$$

Therefore, with the definition of Equation (3.20) it is possible to get the matrix square root from the knowledge of the covariance matrix.

$$\mathbf{S} = \mathbf{V}\begin{pmatrix} \sqrt{d_{11}} & \dots & 0\\ 0 & \ddots & 0\\ 0 & \dots & \sqrt{d_{nn}}\end{pmatrix}\mathbf{V}^{-1} \tag{3.20}$$

So that:

$$\mathbf{SS} = (\mathbf{VD}^{1/2}\mathbf{V}^{-1})(\mathbf{VD}^{1/2}\mathbf{V}^{-1}) = \mathbf{P}_{xx} \tag{3.21}$$

In this thesis the Cholesky Matrix Square Root is adopted. The Cholesky decomposition (see Appendix) asks for a (theoretically) symmetric positive definitive matrix as input for the root evaluation: the result is numerically stable and thus this decomposition is often used in UKF implementation. This formulation can be generalised by exploiting the freedom to choose which of the infinite number of possible square roots to use. The freedom to choose an arbitrary matrix square root comes from the fact that any square root can be found from any other root by applying an orthonormal transformation [26]. If the original matrix is $\mathbf{P}_{xx}$, then a matrix square root $\mathbf{L_1}$ has the following property, which is another definition of the matrix square root.

$$\mathbf{L_1}\mathbf{L_1}^T = \mathbf{P}_{xx} \tag{3.22}$$

It is important to point out that $\mathbf{L_1}$ and $\mathbf{P}_{xx}$ have the same eigenvectors. However, if we define a second matrix square root $\mathbf{L_2} = \mathbf{L_1}\mathbf{U}$ where $\mathbf{U}$ is an orthonormal transformation, then

$$\begin{aligned}\mathbf{L_2}\mathbf{L_2}^T &= (\mathbf{L_1}\mathbf{U})(\mathbf{L_1}\mathbf{U})^T\\ &= \mathbf{L_1}\mathbf{U}\mathbf{U}^T\mathbf{L_1}^T\\ &= \mathbf{L_1}\mathbf{L_1}^T\end{aligned} \tag{3.23}$$

There are no restrictions on using orthogonal or symmetric matrix square roots which are numerically sensitive and computationally expensive to find. Rather, efficient and stable methods such as the Cholesky decomposition can be used, especially for real time estimation where time is one of the main drivers [39] [21].

In the UKF (and hence in the UKFDA) this passage is critical since if something unforeseen happens during the computational iterations of the computer, then the evaluation of matrix **S** is the point where the filter is most likely to have stopped working. Each component of matrix **S** is evaluated separately and a non-positive definitive covariance matrix leads to the root of a negative number, which interrupts the algorithm (see Equation (A.2) in the Appendix).

Once the matrix **S** has been obtained, the sigma points position can be evaluated. The sigma points can (but do not have to) lie on the main axes of $\mathbf{P}_{xx}$.

The weights are different if they are referred to Equation (3.14), i.e. the mean, or Equation (3.15), i.e the covariance. Therefore, there will be two different weights vectors w_m and w_P, one for each moment.

$$w_m{}^{[0]} = \frac{\lambda}{n+\lambda} \tag{3.24}$$

$$w_P{}^{[0]} = w_m{}^{[0]} + (1 - \alpha^2 + \beta) \tag{3.25}$$

$$w_m{}^{[i]} = w_m{}^{[i]} = \frac{1}{2(n+\lambda)} \qquad i = 1, ..., 2n \tag{3.26}$$

Since there is no unique solution, the parameters are free to choose but with some restrictions. The first parameter is $k \geq 0$, and along with $\alpha \in (0, 1]$, it defines how far from the mean the sigma points are located: in each analysis reported on this thesis the values of $k = 3$ and $\alpha = 0.25$ have been selected. Parameter k is positive because a negative choice can lead to a non-positive semi-definite estimate of $\mathbf{P}_{yy}$. Then, there is β which is typically equal to 2 since this value is the optimal choice for Gaussian distributions [39]. The last parameter, λ, is a combination of the others.

$$\lambda = \alpha^2(n + k) - n \tag{3.27}$$

The first and second moment propagation can now be performed following the presented procedure. At first each point is instantiated through the function in Equation (3.1) to yield the set of transformed sigma points.

$$\mathcal{Y}^{[i]} = \mathbf{f}[\mathcal{X}^{[i]}] \tag{3.28}$$

The mean of $\mathbf{y}$ is given by the weighted average of the transformed sigma points.

$$\bar{\mathbf{y}} = \sum_{i=0}^{2n} w_m^{[i]} \mathcal{Y}^{[i]} \tag{3.29}$$

The covariance is the weighted outer product of the transformed sigma points.

$$\mathbf{P}_{yy} = \sum_{i=0}^{2n} w_P^{[i]} (\mathcal{Y}^{[i]} - \bar{\mathbf{y}})(\mathcal{Y}^{[i]} - \bar{\mathbf{y}})^T \tag{3.30}$$

The mean and the covariance of $\mathbf{x}$ are captured precisely up to the second order, therefore the calculated values of mean and covariance of $\mathbf{y}$ are correct to the second order as well, Figure 3.5. This means that the accuracy of the representation of the mean is higher with respect to the linearization case, therefore the UKF will present better performances with respect to the EKF. Moreover, since the UT approximates the distribution of $\mathbf{x}$ rather than $\mathbf{f}[\cdot]$, and thus there is no truncation of the series at a particular order, the unscented algorithm is able to partially incorporate information from the higher orders (see [34] for possible skewness and kurtosis matching).

The UT calculates the first two moments using standard vectors and matrix operations. Consequently the algorithm is suitable for any choice of process model, with a rapid implementation that avoids the evaluation of the Jacobians which are needed in an EKF.

In Figure 3.5, the UT is applied to the sonar example. The results are similar to the DA case which means that the UT correctly approximates moments with the same accuracy. The position of the unscented mean value is the same as the true value (on the figure the two points are one on top the other). Furthermore, the UT is consistent, indeed 1σ covariance contour is only slightly larger.

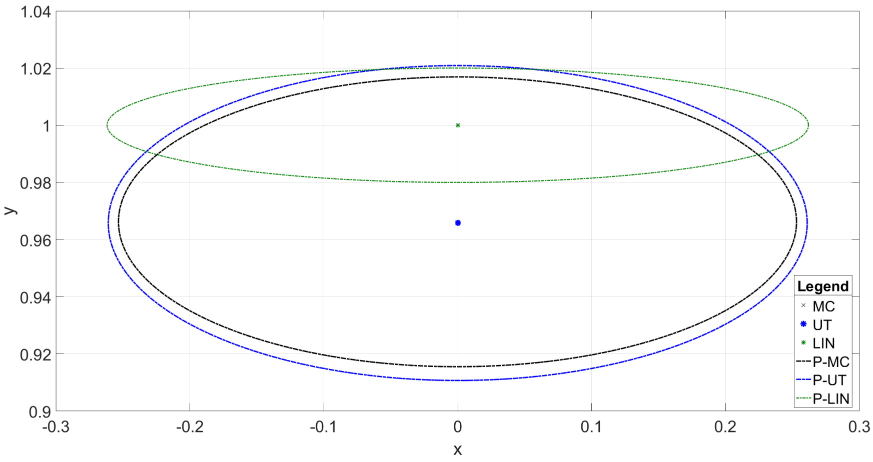

Figure 3.5: Comparison of the mean and covariance computed with the linearization approach and the application of the UT

CHAPTER 4

Nonlinear Filtering

This chapter is devoted to introduce the algorithm of the high-order DA-based extended Kalman filter EKFDA and the algorithm for the classical and DA-based Unscented Kalman filter. A first assessment of the filters performance is provided on an orbit determination problem in Keplerian dynamics.

4.1 Filtering Statement

The equations of motion that describes the dynamical evolution of a spacecraft can be written in the following general form, where the system evolves according to a discrete-time nonlinear state transition equation, Equation (4.1). The only information about this system are its sequence of control inputs and a set of measurements, which are acquired at discrete times. The measurements are related to the state vector by Equation (4.2).

$$\mathbf{x}(k+1) = \mathbf{f}[\mathbf{x}(k), \mathbf{u}(k), k] + \mathbf{v}(k) \tag{4.1}$$

$$\mathbf{z}(k+1) = \mathbf{h}[\mathbf{x}(k), \mathbf{u}(k), k] + \mathbf{w}(k) \tag{4.2}$$

Where $\mathbf{f}[\cdot,\cdot,\cdot]$ is the process model, $\mathbf{x}(k)$ is the m-dimensional state of the spacecraft at time-step k, $\mathbf{u}(k+1)$ is the input vector and $\mathbf{v}(k)$ is the process noise perturbing the spacecraft state, $\mathbf{z}(k+1)$ is the n-dimensional vector of actual measurements at time-step $k+1$, $\mathbf{h}[\cdot,\cdot,\cdot]$ is the measurement function, and $\mathbf{w}(k+1)$ is the measurement noise characterizing the observation error. It is assumed that the process noise vector $\mathbf{v}$ and the measurement noise vector $\mathbf{w}$ are Gaussian and form uncorrelated white sequences, $E[\mathbf{v}(k)] = E[\mathbf{w}(k)] = 0$, with the following autocorrelations for all discrete time-steps i

and j.

$$E[\mathbf{v}(i)\mathbf{v}^T(j)] = \mathbf{Q}(i)\delta_{ij} \qquad \forall\ i,j \tag{4.3}$$

$$E[\mathbf{w}(i)\mathbf{w}^T(j)] = \mathbf{R}(i)\delta_{ij} \qquad \forall\ i,j \tag{4.4}$$

$$E[\mathbf{v}(i)\mathbf{w}^T(j)] = \mathbf{0} \qquad \forall\ i,j \tag{4.5}$$

The Kalman filter propagates the first two moments of the distribution of $\mathbf{x}$ recursively with a distinctive "prediction-correction" structure [36] [11]. Using Julier's and Ulhmann's notation ([21] [22] [19]), let $\hat{\mathbf{x}}(i|j)$ be the estimate of $\mathbf{x}(i)$ using the observation information up to an including time j, and $\mathbf{Z}^j = [\mathbf{z}(1),\ldots,\mathbf{z}(j)]$. The conditional covariance of the estimate is $\mathbf{P}(i|j)$. Given an estimate of the spacecraft state $\hat{\mathbf{x}}(k|k)$, the filter predicts what the future state will be using the process model. The predicted quantities are so calculated.

$$\hat{\mathbf{x}}(k+1|k) = E\big[\mathbf{f}[\mathbf{x}(k),\mathbf{u}(k),k] + \mathbf{v}(k)|\mathbf{Z}^k\big] \tag{4.6}$$

$$\mathbf{P}(k+1|k) = E\big[\{\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|k)\}\{\mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1|k)\}^T|\mathbf{Z}^k\big] \tag{4.7}$$

Generally $\mathbf{f}[\cdot]$ and $\mathbf{h}[\cdot]$ are nonlinear, hence they challenge the precise propagation of the statistics of $\mathbf{x}(k)$. In this thesis it is assumed that the distribution of $\mathbf{x}(k)$ is Gaussian and completely parametrised by just the mean and the covariance.

First, the current state and the associated covariance are transformed through the state transition and the observation equations. These values $\hat{\mathbf{x}}(k+1|k)$ and $\mathbf{P}(k+1|k)$ are the *predicted* quantities since they represent the estimation of the state of the system at time $k+1$ given the knowledge of all observations up to time k. In the second step, the *update*, the system is observed at time $k+1$ and the filter uses the observation information (the measurements) to correct the predicted quantities, producing the estimate $\hat{\mathbf{x}}(k+1|k+1)$ and $\mathbf{P}(k+1|k+1)$.

As far as the two DA-based filters introduced in this work are concerned, they both use the update equations of a classical Kalman filter. They differ instead by the way the prediction step is performed. The update equations are:

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)n(k+1) \tag{4.8}$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{K}(k+1)\mathbf{P}_{zz}(k+1)\mathbf{K}^T(k+1) \tag{4.9}$$

Where vector $n(k+1)$ is the innovation, which is evaluated as the difference between the actual observation at time k, $\mathbf{z}(k+1)$, and the predicted mean observation $\hat{\mathbf{z}}(k+1|k)$.

$$n(k+1) = \mathbf{z}(k+1) - \hat{\mathbf{z}}(k+1|k) \tag{4.10}$$

Furthermore, $\mathbf{K}(k+1)$ is the Kalman gain matrix, evaluated with the information of the covariance matrix of the state measurements $\mathbf{P}_{zz}(k+1|k)$ and the cross-covariance matrix of the state and the measurement $\mathbf{P}_{xz}(k+1|k)$.

$$\mathbf{K}(k+1) = \mathbf{P}_{xz}(k+1|k)\mathbf{P}_{zz}^{-1}(k+1|k) \tag{4.11}$$

It can be seen that the *update* equations are only a function of the predicted values of the first two moments of $\mathbf{x}(k)$ and $\mathbf{z}(k)$.

4.2 High Order Extended Kalman Filter

The high-order DA-based extended Kalman filter, here called EKFDA, uses the representation of quantities such as the state vector and the measurement vector in their Taylor series expansion to perform the *prediction* step [16]. The arbitrary choice of the filter order let the EKFDA work as a simple EKF in the case of order 1. The prediction step works on the current state and related covariance to estimate at time $k{+}1$ the mean and the covariance of the state vector, $\hat{\mathbf{x}}(k+1|k)$ and $\mathbf{P}(k+1|k)$, and the mean of the measurements, $\hat{\mathbf{z}}(k+1|k)$.

$$\hat{\mathbf{x}}_i(k+1|k) = E\big[\mathbf{f}_i[\mathbf{x}(k), \mathbf{u}(k), k] + \mathbf{v}_i(k)\big] \tag{4.12}$$

$$\begin{aligned}\mathbf{P}_{ij}(k+1|k) = E\big[&\{\mathbf{f}_i[\mathbf{x}(k), \mathbf{u}(k), k] - \hat{\mathbf{x}}_i(k+1|k) + \mathbf{v}_i(k)\}\cdot \\ &\cdot\{\mathbf{f}_j[\mathbf{x}(k), \mathbf{u}(k), k] - \hat{\mathbf{x}}_j(k+1|k) + \mathbf{v}_j(k)\}\big]\end{aligned} \tag{4.13}$$

$$\hat{\mathbf{z}}_p(k+1|k) = E\big[\mathbf{h}_p[\mathbf{x}(k), \mathbf{u}(k), k] + \mathbf{w}_p(k)\big] \tag{4.14}$$

These equations, which recall Equations (4.6) and (4.7), are referred to the singular vector component, where $i, j = 1, \ldots, m$ and $p = 1. \ldots, n$. The prediction step, as it has been defined in this thesis, includes also the evaluation of the cross-covariance matrix of the state and the measurement, and the covariance matrix of the measurement ($q = 1. \ldots, n$).

$$\begin{aligned}\mathbf{P}_{zz,pq}(k+1|k) = E\big[&\{\mathbf{h}_p[\mathbf{x}(k), \mathbf{u}(k), k] - \hat{\mathbf{z}}_p(k+1|k) + \mathbf{w}_p(k)\}\cdot \\ &\cdot\{\mathbf{h}_q[\mathbf{x}(k), \mathbf{u}(k), k] - \hat{\mathbf{z}}_q(k+1|k) + \mathbf{w}_q(k)\}\big]\end{aligned} \tag{4.15}$$

$$\begin{aligned}\mathbf{P}_{xz,ip}(k+1|k) = E\big[&\{\mathbf{f}_i[\mathbf{x}(k), \mathbf{u}(k), k] - \hat{\mathbf{x}}_i(k+1|k) + \mathbf{v}_i(k)\}\cdot \\ &\cdot\{\mathbf{h}_p[\mathbf{x}(k), \mathbf{u}(k), k] - \hat{\mathbf{z}}_p(k+1|k) + \mathbf{w}_p(k)\}\big]\end{aligned} \tag{4.16}$$

In the classical EKF, the Taylor series expansions of the equations of motion (4.1) and measurement (4.2) are linearized about the current estimate of the mean:

$$\mathbf{x}(k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{A}(k)\delta\mathbf{x}(k) + \mathbf{w}(k) \tag{4.17}$$

$$\mathbf{z}(k+1) = \mathbf{h}[\hat{\mathbf{x}}_i(k+1|k), \mathbf{u}(k), k] + \mathbf{C}(k)\delta\mathbf{x}(k) + \mathbf{w}(k) \tag{4.18}$$

where $\delta\mathbf{x}(k)$ is deviation of the true trajectory from the estimated mean.

$$\delta\mathbf{x}(k) = \mathbf{x}(k) - \hat{\mathbf{x}}(k|k) \tag{4.19}$$

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{f}[\hat{\mathbf{x}}(k|k), \mathbf{u}(k), k] \tag{4.20}$$

Therefore, the equations of the prediction step can take advantage of the linearity with respect to the state.

DA, on the other hand, can easily provide the Taylor series expansion up to the selected order, both for $\mathbf{f}[\cdot,\cdot,\cdot]$ and $\mathbf{h}[\cdot,\cdot,\cdot]$. Thus, DA can generalize Equation (4.17) and (4.18) into the arbitrary order form.

$$\begin{aligned} x_i(k+1) = {} & \mathbf{f}_i[\hat{\mathbf{x}}(k|k), \mathbf{u}(k), k] + \\ & + \sum_{r=1}^{c} \frac{1}{r!} \mathbf{f}^{\gamma_1 \ldots \gamma_r}_{i,(k+1|k)} \delta x_1^{\gamma_1}(k) \ldots \delta x_m^{\gamma_r}(k) + \mathbf{v}_i(k) \end{aligned} \tag{4.21}$$

$$\begin{aligned} z_p(k+1) = {} & \mathbf{h}_p\big[\mathbf{f}[\hat{\mathbf{x}}(k|k), \mathbf{u}(k), k], \mathbf{u}(k), k\big] + \sum_{r=1}^{c} \frac{1}{r!} \mathbf{h}^{\gamma_1 \ldots \gamma_r}_{p,(k+1|k)} \cdot \\ & \cdot \delta x_1^{\gamma_1}(k) \ldots \delta x_m^{\gamma_r}(k) + \mathbf{w}_i(k) \end{aligned} \tag{4.22}$$

Where c is the order of the expansion and $\gamma_i \in \{1, \ldots, m\}$. Furthermore, $\mathbf{f}^{\gamma_1 \ldots \gamma_r}_{i,(k+1|k)}$ includes the higher-order partials of the solution flow, which maps the deviations from time k to time $k+1$ and, in an analogue way, $\mathbf{h}^{\gamma_1 \ldots \gamma_r}_{p,(k+1|k)}$ includes the higher-order partials of the measurement function. These two functions are both obtained by integration in the DA framework of the equations of motion and evaluating the measurement equation [16] [29] .

The two expansions (4.21) and (4.22) can be inserted into the prediction equations ((4.12), (4.13), (4.14), (4.15), (4.16)) to predict the means and covariances.

$$\begin{aligned}\hat{\mathbf{x}}_i(k+1|k) = &\ \mathbf{f}_i[\hat{\mathbf{x}}(k|k), \mathbf{u}(k), k] + \\ &+ \sum_{r=1}^{c} \frac{1}{r!} \mathbf{f}^{\gamma_1 \dots \gamma_r}_{i,(k+1|k)} E[\delta x_1^{\gamma_1}(k) \dots \delta x_m^{\gamma_r}(k)]\end{aligned} \tag{4.23}$$

$$\begin{aligned}\mathbf{P}_{ij}(k+1|k) = &\sum_{r=1}^{c}\sum_{s=1}^{c} \frac{1}{r!s!} \mathbf{f}^{\gamma_1 \dots \gamma_r}_{i,(k+1|k)} \mathbf{f}^{\xi_1 \dots \xi_s}_{j,(k+1|k)} \cdot \\ &\cdot E[\delta x_1^{\gamma_1}(k) \dots \delta x_m^{\gamma_r}(k) \delta x_1^{\xi_1}(k) \dots \delta x_m^{\xi_s}(k)] + \\ &- \delta\hat{x}_i(k+1)\delta\hat{x}_j(k+1) + \mathbf{Q}_{ij}(k)\end{aligned} \tag{4.24}$$

$$\begin{aligned}\hat{\mathbf{z}}_p(k+1|k) = &\ \mathbf{h}_p\big[\mathbf{f}[\hat{\mathbf{x}}(k|k), \mathbf{u}(k), k], \mathbf{u}(k), k\big] + \\ &+ \sum_{r=1}^{c} \frac{1}{r!} \mathbf{h}^{\gamma_1 \dots \gamma_r}_{p,(k+1|k)} E[\delta x_1^{\gamma_1}(k) \dots \delta x_m^{\gamma_r}(k)]\end{aligned} \tag{4.25}$$

$$\begin{aligned}\mathbf{P}_{zz,pq}(k+1|k) = &\sum_{r=1}^{c}\sum_{s=1}^{c} \frac{1}{r!s!} \mathbf{h}^{\gamma_1 \dots \gamma_r}_{p,(k+1|k)} \mathbf{h}^{\xi_1 \dots \xi_s}_{q,(k+1|k)} \cdot \\ &\cdot E[\delta x_1^{\gamma_1}(k) \dots \delta x_m^{\gamma_r}(k) \delta x_1^{\xi_1}(k) \dots \delta x_m^{\xi_s}(k)] + \\ &- \delta\hat{z}_p(k+1)\delta\hat{z}_q(k+1) + \mathbf{R}_{pq}(k+1)\end{aligned} \tag{4.26}$$

$$\begin{aligned}\mathbf{P}_{xz,ip}(k+1|k) = &\sum_{r=1}^{c}\sum_{s=1}^{c} \frac{1}{r!s!} \mathbf{f}^{\gamma_1 \dots \gamma_r}_{i,(k+1|k)} \mathbf{h}^{\xi_1 \dots \xi_s}_{p,(k+1|k)} \cdot \\ &\cdot E[\delta x_1^{\gamma_1}(k) \dots \delta x_m^{\gamma_r}(k) \delta x_1^{\xi_1}(k) \dots \delta x_m^{\xi_s}(k)] + \\ &- \delta\hat{x}_i(k+1)\delta\hat{z}_p(k+1)\end{aligned} \tag{4.27}$$

Where $\xi_i \in \{1, \dots, m\}$; $\delta\hat{x}_i(k+1) = \mathbf{f}_i[\hat{\mathbf{x}}(k|k), \mathbf{u}(k), k] - \hat{\mathbf{x}}_i(k+1|k)$; and $\delta\hat{z}_p(k+1) = \mathbf{h}_p\big[\mathbf{f}[\hat{\mathbf{x}}(k|k), \mathbf{u}(k), k], \mathbf{u}(k), k\big] - \hat{\mathbf{z}}_p(k+1|k)$. Since the distribution of the state has been assumed to be Gaussian, the moments can be completely described by the first two moments and can be easily computed in terms of covariance matrix using Isserlis' formula on the monomials of the Taylor polynomial.

Once the prediction step has been performed, the update step can be carried out as described in the previous section.

4.3 Unscented Kalman Filter

The Unscented Kalman Filter UKF relies its prediction step on the unscented transformation. The state vector is often augmented with the process noise term in order to increase the filter robustness, since the noise terms have well known moments with null mean (Gaussian noise) [45]. This option leads to the creation of additional sigma points, which means that the effects of the process noise are introduced with the same order of accuracy as the uncertainty in the state.
The UKF prediction starts with the set of sigma points, which is created applying Equations (3.16), (3.17), and (3.18) to the given system, and their relative weights, Equations (3.24), (3.25), and (3.26). The set gets transformed by initializing each sigma point through the process model, giving the transformed set.

$$\mathcal{X}^{[i]}(k+1|k) = \mathbf{f}[\mathcal{X}^{[i]}(k|k), \mathbf{u}(k), k] \tag{4.28}$$

The mean and the covariance of the propagated set is evaluated by weighting each transferred sigma point.

$$\hat{\mathbf{x}}(k+1|k) = \sum_{i=0}^{2m} w_m^{[i]} \mathcal{X}^{[i]}(k+1|k) \tag{4.29}$$

$$\begin{aligned}\mathbf{P}(k+1|k) = \sum_{i=0}^{2m} w_P^{[i]} \{\mathcal{X}^{[i]}(k+1|k) - \hat{\mathbf{x}}(k+1|k)\}\cdot \\ \cdot \{\mathcal{X}^{[i]}(k+1|k) - \hat{\mathbf{x}}(k+1|k)\}^T\end{aligned} \tag{4.30}$$

The same line of thinking is adopted for the measurement vector, where each one of the prediction sigma points goes through the observation model, and then the predicted measurements are calculated.

$$\mathcal{Z}^{[i]}(k+1|k) = \mathbf{h}[\mathcal{X}^{[i]}(k+1|k), \mathbf{u}(k), k] \tag{4.31}$$

$$\hat{\mathbf{z}}(k+1|k) = \sum_{i=0}^{2m} w_m^{[i]} \mathcal{Z}^{[i]}(k+1|k) \tag{4.32}$$

And, since the observation noise is independent and additive, the covariance matrix of the measurement and the cross-covariance matrix of the state and the measurement are so evaluated.

$$\mathbf{P}_{zz}(k+1|k) = \sum_{i=0}^{2m} w_P^{[i]} \{\mathcal{Z}^{[i]}(k+1|k) - \hat{\mathbf{z}}(k+1|k)\}\cdot$$
$$\cdot \{\mathcal{Z}^{[i]}(k+1|k) - \hat{\mathbf{z}}(k+1|k)\}^T + \mathbf{R}(k+1) \tag{4.33}$$

$$\mathbf{P}_{xz}(k+1|k) = \sum_{i=0}^{2m} w_P^{[i]} \{\mathcal{X}^{[i]}(k+1|k) - \hat{\mathbf{x}}(k+1|k)\}\cdot$$
$$\cdot \{\mathcal{Z}^{[i]}(k+1|k) - \hat{\mathbf{z}}(k+1|k)\}^T \tag{4.34}$$

The UKF has completed its prediction process and can now use the update equations for the correction part of the Kalman filters.

4.4 DA-based Unscented Kalman Filter

The UKF can be modified into the DA-based Unscented Kalman Filter, the UKFDA. This filter provides an improvement by reducing the computational time of the classical UKF when the equations of motion are particularly complex. DA is used to Taylor expand the function f in Equation (4.20). As a result, it builds an analytical map that connects the state at time k with the state at time $k+1$. The resulting polynomials can be evaluated to map the sigma points through the model equations, by replacing then the evaluations of f. Consequently, the DA-based approach tends to outperform the classical one when the evaluation of f and h is computationally demanding. The order at which the Taylor polynomial is computed can be arbitrarily selected.

More specifically, at each step, the state $\mathbf{x}$ is initialized as DA variable around the current mean and propagated in the DA framework through the equations of motions and of the measurements. Then the polynomials are evaluated at the sigma points. The distance of each single sigma point is known and given by the columns (or rows) of matrix $\mathbf{S} = \pm\sqrt{(n+\lambda)\mathbf{P}}$ (see Equations (3.17) and (3.18)). The sigma points are propagated by simply evaluating the Taylor expansion at each column of $\mathbf{S}$. As a results, the UKFDA presents faster performances with respects to the UKF because f is evaluated only once in the DA framework.

However, the introduction of the DA in the filter is useful only if the problem has a certain level of complexity. If the process model $\mathbf{f}[\cdot,\cdot,\cdot]$ is not numerically complex, then the creation of the polynomial and the relative sigma point evaluation could take longer than the direct propagation of the $2n+1$ sigma points.

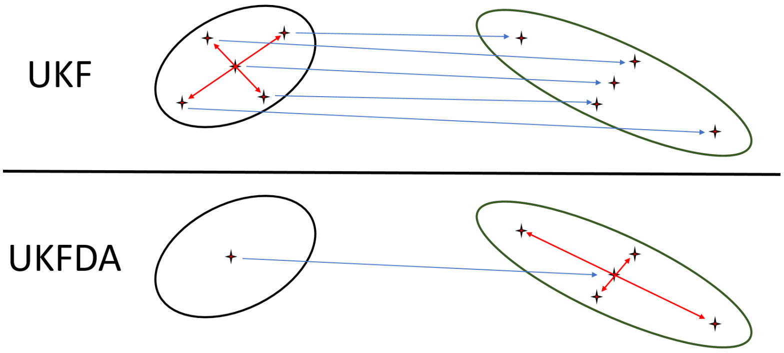

Figure 4.1: The propagation of the sigma points in the UKF and UKFDA approaches.

Figure 4.1 gives a visual idea of the different propagation technique used by the two unscented Kalman filters.

4.5 Orbit Determination in Two-Body Dynamics

The performance of the different filters is here assessed on a test case: the determination of the orbit of a spacecraft around the Earth. The equations of motion governing the system are the ones associated to Keplerian dynamics, where $\mathbf{r}$ is the position vector of the spacecraft and μ is the Earth gravitational parameter.

$$\frac{d\dot{\mathbf{r}}}{dt} = -\frac{\mu}{r^3}\mathbf{r} \tag{4.35}$$

The problem is taken from [16]. The initial true position and velocity assumed for the test are reported in Equation (4.36), where each length unit has been scaled by the orbit semi-major axis, $a = 8788\ km$ and by the time $\sqrt{\dfrac{a^3}{\mu}}$.

$$\mathbf{x_0} = \begin{pmatrix} \mathbf{r_0} \\ \mathbf{v_0} \end{pmatrix} = \begin{pmatrix} -0.68787 \\ -0.39713 \\ +0.28448 \\ -0.51331 \\ +0.98266 \\ +0.37611 \end{pmatrix} \tag{4.36}$$

The initial estimates for the state have a 10% offset from the true initial state values expressed in Equation (4.36). The initial error covariance matrix has been assumed as diagonal, the value of the variance for the position vector components is 0.01, while the variance for the velocity vector components is 10^{-4}.

The measurement model assumes the radial position of the spacecraft with respect to the Earth and the line of sight direction of the planet.

$$z_1 = r + w_1 \tag{4.37}$$

$$z_2 = \arctan\left(\frac{y}{x}\right) + w_2 \tag{4.38}$$

$$z_3 = \arcsin\left(\frac{z}{r}\right) + w_3 \tag{4.39}$$

Where w_i, with $i = 1, 2, 3$, represents the related measurement noise component. This simulation provides no process noise, while the standard deviation of the error is assumed to be 10^{-3} for the radial position, and $1.745 \cdot 10^{-6}$ for the angle error, following the conventional measurement accuracy for the directional measurement.

In the results that are going to be shown, different filter orders are expressed in the filters names with a number. Thus, for example, UKFDA2 indicates that the DA-based Unscented Kalman Filter has been used with a second order polynomial.

Figures 4.2 and 4.3 show the comparison of the position and error profiles obtained with the EKFDA1, EKFDA2, EKFDA3 and UKFDA2. The position and velocity errors, expressed respectively as ϵ_r and ϵ_v, are defined as the Euclidean norm of the difference vector between the estimated position and velocity with the corresponding true values. The two figures represent the time duration of two orbits, with a total of 12 observations each orbit separated by the same time interval. The EKFDA1 line is equivalent to the performance of a classical EKF, therefore it is possible to directly compare both the UKFDA and EKFDA with the classical EKF. The simulation results show how the second order filter significantly improves the estimation accuracy, proving that higher order filters can extract more information from the available nonlinear equations with respect to the first order. However, there appears to be no benefits into upgrading to third order from second order. This lack of improvement is connected to the decision, that is intrinsic in the Kalman approach, of describing the propagated statistics using only the first and second order moments (see [16]).

The filters based on the unscented transformation show same accuracy level of the second order DA-based extended Kalman filter. As already stated

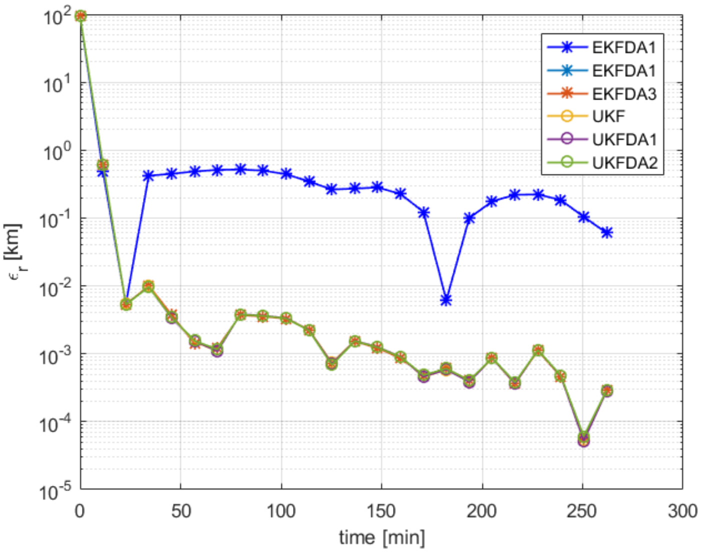

Figure 4.2: Orbit determination test, 12 observations per orbit: position error profiles of the EKFDA1, EKFDA2, EKFDA3, UKF, UKFDA1 and UKFDA2.

when talking about UT, the UKF filter type can precisely predict the first two moments of a variable after propagation, thus UKFDA1, UKFDA2 and UKF accuracy is comparable to the EKFDA2.

The difference between the DA-based and the standard UKF resides mostly on the computational time and can't be appreciated in a fast application such as the Keplerian dynamics. A more detailed time performance analysis has been performed for the relative pose application in the next chapter.

UKFDA1 and UKFDA2 presents no valuable difference in this application since the second order terms of the polynomials are relatively small.

Figure 4.4 represents the standard deviation profiles both for the spacecraft position and velocity. These quantities are calculated from the diagonal terms of the estimate position and velocity covariance matrices: σ_{rx}^2, σ_{ry}^2, σ_{rz}^2, σ_{vx}^2, σ_{vy}^2, σ_{vz}^2.

$$\sigma_r = \sqrt{\sigma_{rx}^2 + \sigma_{ry}^2 + \sigma_{rz}^2} \tag{4.40}$$

$$\sigma_v = \sqrt{\sigma_{vx}^2 + \sigma_{vy}^2 + \sigma_{vz}^2} \tag{4.41}$$

The filters give the same behaviour when the measurements frequency is changed. Figure 4.6 and 4.7 show the position error profile accuracy for

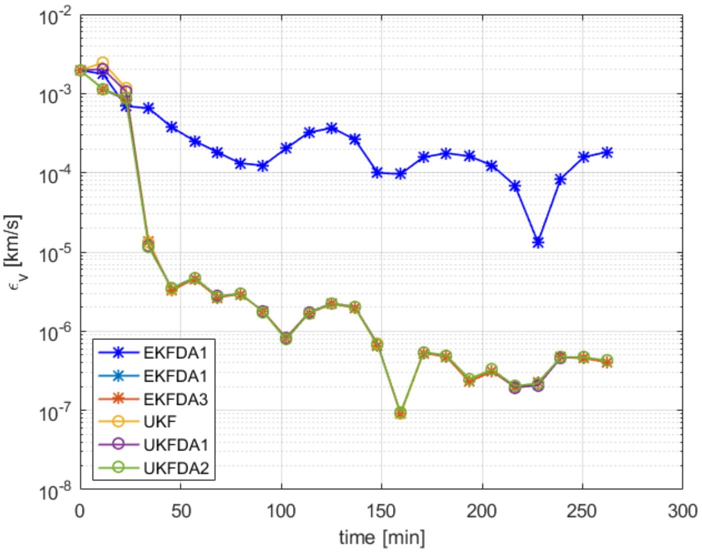

Figure 4.3: Orbit determination test, 12 observations per orbit: velocity error profiles of the EKFDA1, EKFDA2, EKFDA3, UKF, UKFDA1 and UKFDA2.

different observation frequencies, respectively 6 and 24 measurements per orbit. As in the case of 12 observations, moving from the first order EKF to EKFDA2 or to an unscented filter improves significantly the estimation accuracy.

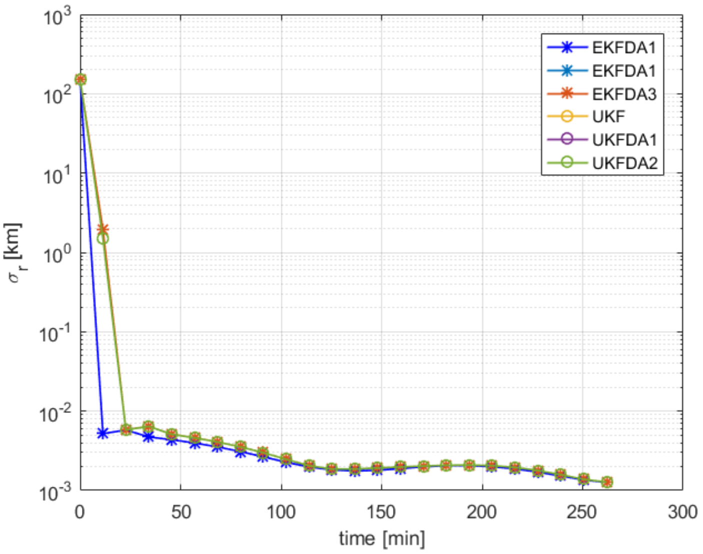

Figure 4.4: Orbit determination test, 12 observations per orbit: σ_r profile of the EKFDA1, EKFDA2, EKFDA3, UKF, UKFDA1 and UKFDA2.

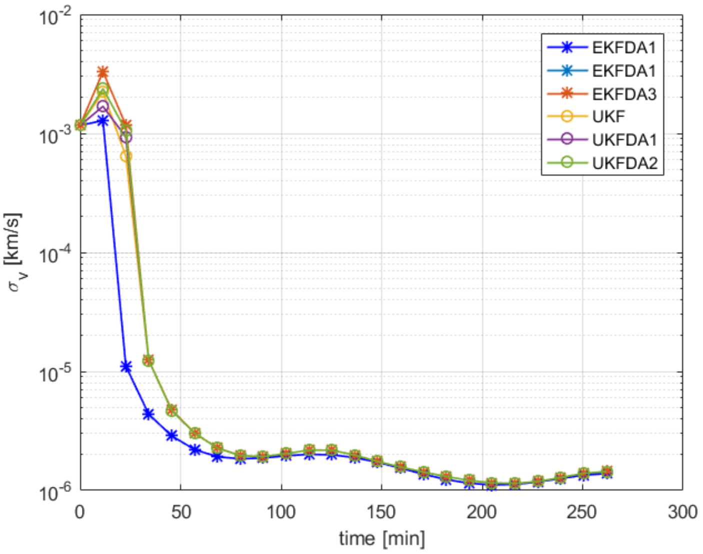

Figure 4.5: Orbit determination test, 12 observations per orbit: σ_v profile of the EKFDA1, EKFDA2, EKFDA3, UKF, UKFDA1 and UKFDA2.

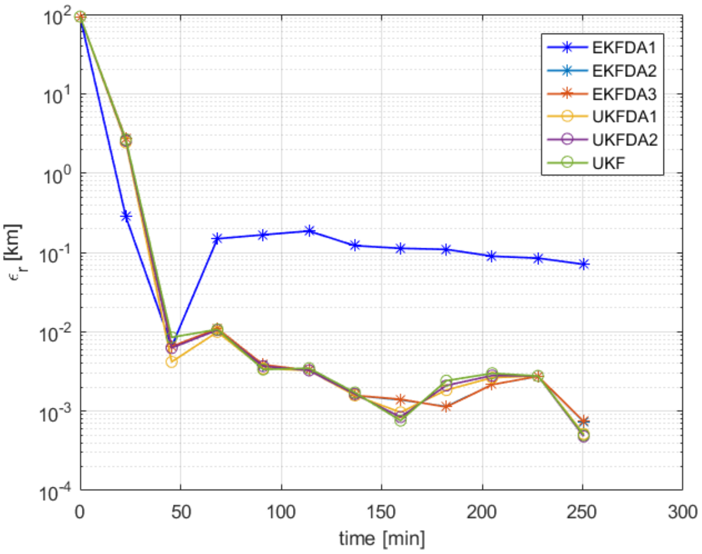

Figure 4.6: Orbit determination test, 6 observations per orbit: ϵ_r profile of the EKFDA1, EKFDA2, EKFDA3, UKF, UKFDA1 and UKFDA2.

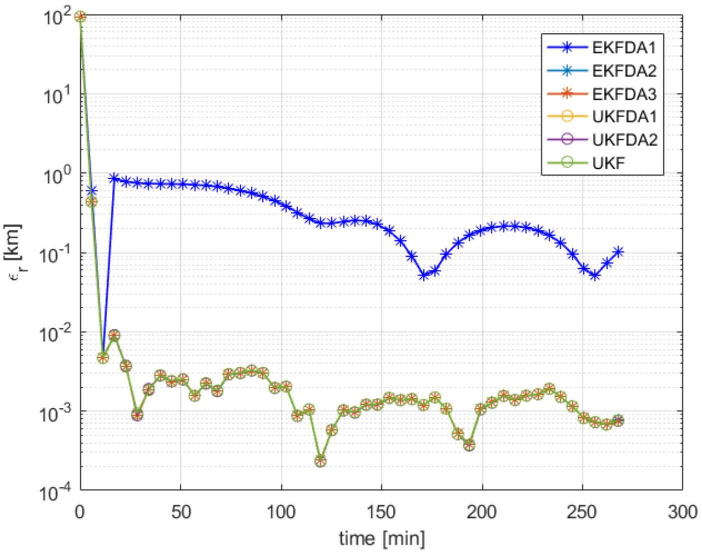

Figure 4.7: Orbit determination test, 24 observations per orbit: ϵ_r profile of the EKFDA1, EKFDA2, EKFDA3, UKF, UKFDA1 and UKFDA2.

CHAPTER 5

Application: ENVISAT Relative Pose Estimation

5.1 Problem Statement

Nowadays more than ever space pollution is a relevant problem. Active debris removal is just one of the many application filters are used in: they are a key component for navigation systems, guidance, but mostly rendezvous problems, especially if the target is uncooperative. In order to solve estimation problems such as localization, mapping, and tracking, the filter must determine the relative position and orientation (pose) between the chaser and the target [37] [14] [25].

The chaser is approaching the uncooperative target to perform the docking maneuver to remove it. Thus, the chaser needs to estimate the relative pose of the target and the filter is the software applied for this task: it takes the target images from a camera (as input) and informs the chaser about its relative pose with the target (as output).

The thesis applies the developed filters to the problem of docking and deorbiting ENVISAT, following the effort of the European Space Agency. [9].

The analysis is performed by making the following assumptions [16]:

- *a priori* knowledge of the geometrical and physical characteristics of both chaser and target [1]
- chaser motion supposed to be deterministic
- rigid-body dynamics
- no external disturbances

- no control actions

It must be noticed that neglecting external disturbances and flexibility implies the possibility of completely decouple the translational problem and the rotational one. However, as explained later in the chapter, the adopted measurement model asks for the simultaneous propagation of both the translational and rotational dynamics.

5.2 Dynamics

5.2.1 Translational Dynamics

Absolute Chaser Motion

The motion of the chaser is described by the following equations, where μ is the gravitational parameter of the Earth, $\bar{r}$ is the position of the chaser centre of mass with respect to the Earth, and θ indicates the true anomaly in the orbit of the chaser.

$$\ddot{\bar{r}} = \bar{r}\dot{\theta}^2 - \frac{\mu}{\bar{r}^2} \tag{5.1}$$

$$\ddot{\theta} = -2\frac{\dot{\bar{r}}\dot{\theta}}{\bar{r}} \tag{5.2}$$

Relative Translational Dynamics

The target, ENVISAT, has its relative translational dynamic equations developed with respect to the chaser local-vertical-local-horizontal (LVLH) frame of the chaser. The target relative position, denoted as $\mathbf{r_r}$, and relative velocity, $\mathbf{v_r}$, are defined in the chaser LVLH frame as expressed in Equations (5.3) and (5.4).

$$\mathbf{r_r} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}} \tag{5.3}$$

$$\mathbf{v_r} = \dot{x}\hat{\mathbf{i}} + \dot{y}\hat{\mathbf{j}} + \dot{z}\hat{\mathbf{k}} \tag{5.4}$$

Where x, y, and z represent the three components of vector $\mathbf{r_r}$ in the chaser LVLH frame and $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ are the corresponding unit vectors of the reference frame. Thus, the equations of motion of the target for its relative translational dynamics can be written in the following way.

$$\ddot{x} = +2\dot{\theta}\dot{y} + \ddot{\theta}y + \dot{\theta}^2 x - \frac{\mu(\ddot{r} + x)}{[(\bar{r} + x)^2 + y^2 + z^2]^{3/2}} + \frac{\mu}{\bar{r}^2} \tag{5.5}$$

$$\ddot{y} = -2\dot{\theta}\dot{x} - \ddot{\theta}x + \dot{\theta}^2 y - \frac{\mu y}{[(\bar{r} + x)^2 + y^2 + z^2]^{3/2}} \tag{5.6}$$

$$\ddot{z} = -\frac{\mu z}{[(\bar{r} + x)^2 + y^2 + z^2]^{3/2}} \tag{5.7}$$

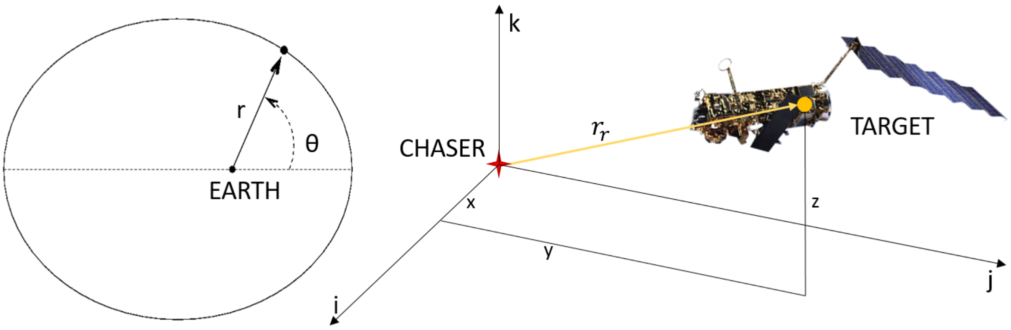

Figure 5.1: Translational dynamics elements representation.

5.2.2 Rotational Dynamics

Similar to the translational dynamics, the rotational dynamics of the target relies on the problem setted in [16]. The relative orientation of the body-fixed frame on the target with respect to the body-fixed frame of the chaser can be described through a rotational matrix $\mathbf{\Gamma}$. Therefore, the relative angular velocity $\boldsymbol{\omega_r}$ in the target body-fixed reference frame is function of the angular velocity of the chaser $\boldsymbol{\omega_c}$ and the angular velocity of the target $\boldsymbol{\omega_t}$, both expressed in their own body-fixed reference frames.

$$\boldsymbol{\omega_r} = \boldsymbol{\omega_t} - \mathbf{\Gamma}\boldsymbol{\omega_c} \tag{5.8}$$

$$\boldsymbol{\dot{\omega}_r} = \boldsymbol{\dot{\omega}_t} - \mathbf{\Gamma}\boldsymbol{\dot{\omega}_c} + \boldsymbol{\omega_r} \wedge \mathbf{\Gamma}\boldsymbol{\omega_c} \tag{5.9}$$

The relative attitude of the target is determined by the parametrization of rotation matrix $\mathbf{\Gamma}$. The Modified Rodriguez Parameters, MRP [13] [48] [12], are adopted as the minimal set of three parameters that allows to overcome singularities and to describe every rotation with the minor angle definition. This parametrization is derived by applying a stereographic projection of the quaternions. The quaternions $\mathbf{q}$ are expressed as follows.

$$\mathbf{q} = \begin{bmatrix} \bar{\mathbf{q}} \\ q_4 \end{bmatrix} \tag{5.10}$$

$$\bar{\mathbf{q}} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \hat{\mathbf{n}} \sin\left(\frac{\phi}{2}\right) \tag{5.11}$$

$$q_4 = \cos\left(\frac{\phi}{2}\right) \tag{5.12}$$

Where $\hat{\mathbf{n}}$ is a unit vector corresponding to the axis of rotation and ϕ is the angle of rotation. The MRP are related to the quaternions in the following way.

$$\mathbf{p} = \frac{\bar{\mathbf{q}}}{(1+q_4)} = \hat{\mathbf{n}} \tan\left(\frac{\phi}{4}\right) \tag{5.13}$$

Vector $\mathbf{p}$ is the MRP vector, with dimension 3x1. The kinematic equation of motion are derived by using the target's relative angular velocity, therefore the time evolution of the MRP is described by Equation (5.14).

$$\dot{\mathbf{p}} = \frac{1}{4}\left[(1 - \mathbf{p}^T\mathbf{p})\mathbf{I_3} + 2\mathbf{p}\mathbf{p}^T + 2[\mathbf{p}\wedge]\right]\boldsymbol{\omega_r} \tag{5.14}$$

Where $\mathbf{I_3}$ is a 3 x 3 identity matrix and $[\mathbf{p}\wedge]$ is a 3 x 3 cross product matrix defined as:

$$[\mathbf{p}\wedge] = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_2 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix} \tag{5.15}$$

The rotation matrix that connects the chaser body-fixed frame and the target body-fixed frame can thus be evaluated.

$$\begin{cases} \alpha_1 = 4\dfrac{1-\mathbf{p}^T\mathbf{p}}{(1+\mathbf{p}^T\mathbf{p})^2} \\ \\ \alpha_2 = 8\dfrac{1}{(1+\mathbf{p}^T\mathbf{p})^2} \end{cases} \tag{5.16}$$

$$\boldsymbol{\Gamma}(\mathbf{p}) = \mathbf{I_3} - \alpha_1[\mathbf{p}\wedge] + \alpha_2[\mathbf{p}\wedge]^2 \tag{5.17}$$

The absolute rotational dynamics of the chaser is described by the torque-free Euler equations.

The relative attitude dynamics are obtained by substituting the kinematics relationship in the Euler absolute equations of the target spacecraft.

$$\mathbf{J_t}\dot{\boldsymbol{\omega}}_{\boldsymbol{r}} + \boldsymbol{\omega}_{\boldsymbol{r}} \wedge \mathbf{J_t}\boldsymbol{\omega}_{\boldsymbol{r}} = \mathbf{M_{app}} - \mathbf{M_g} - \mathbf{M_{ci}} \tag{5.18}$$

Where $\mathbf{J_t}$ is the matrix of inertia of the target, $\mathbf{M_{app}}$ is the apparent torques, $\mathbf{M_g}$ is the gyroscopic torques, and $\mathbf{M_{ci}}$ is the chaser-inertial torques.

$$\mathbf{M_{app}} = \mathbf{J_t}\boldsymbol{\omega}_{\boldsymbol{r}} \wedge \boldsymbol{\Gamma}\boldsymbol{\omega}_{\boldsymbol{c}} \tag{5.19}$$

$$\mathbf{M_g} = \boldsymbol{\Gamma}\boldsymbol{\omega}_{\boldsymbol{c}} \wedge \mathbf{J_t}\boldsymbol{\Gamma}\boldsymbol{\omega}_{\boldsymbol{c}} + \boldsymbol{\omega}_{\boldsymbol{r}} \wedge \mathbf{J_t}\boldsymbol{\omega}_{\boldsymbol{c}} + \boldsymbol{\Gamma}\boldsymbol{\omega}_{\boldsymbol{c}} \wedge \mathbf{J_t}\boldsymbol{\omega}_{\boldsymbol{r}} \tag{5.20}$$

$$\mathbf{M_{ci}} = \mathbf{J_t}\boldsymbol{\Gamma}\dot{\boldsymbol{\omega}}_{\boldsymbol{c}} \tag{5.21}$$

5.3 Measurement Model

5.3.1 Markers Creation

Filters need reliable measurements to perform a good correction on the predicted quantities to properly determine the target attitude. The majority of filters used for space applications rely on camera image processing. In real applications, the image processing software is set up to look for target points in each image taken: these points are refereed to as markers [47]. The software processes the image sent from the camera and analyses it: once the position of the markers is found, it sends this information to the filter. The marker selection is a complex trade-off based on the target shape, volume and colour since the processing of the image needs to be as fast as possible. A common solution is to select the target corners as markers and to base the processing software on one of the most reliable corner detecting algorithms, such as the Harris-Stephens algorithm [52] and the Förstner algorithm.

Regardless the selected algorithm, there is the need of an effective interaction between the filter and the image processing software. After a transient time, where the first measurements are made and the position error by the filter estimation rapidly drops, the communication between the image processing software and the filter should be optimized to achieve a faster marker estimation. When the filter ends its iterative cycle, it can directly tell the camera where to look for the markers in the next image. In this way, the camera software can analyse a smaller region of the image and, thus, it will not have to process all the image pixels, but just the ones that are nearby the position indicated by the filter.

Back to the ENVISAT relative pose estimation problem, it has been decided to select the corners of the main body as markers and to track their position in time. Figure 5.2 shows the dimension of the spacecraft and [1] gives information about ENVISAT's mass, its centre of mass location (with no propellant), its moments of inertia (with no propellant) and its geometrical centre and volume. The marker positions can therefore provide information about the spacecraft attitude since the position of each marker is well known with respect to the centre of mass. By tracking the marker trajectory, the filter is able to reconstruct the state of the spacecraft and calculate relative position and velocity. ENVISAT main body, without the solar panel, can be described as a simple parallelepiped with 8 corners: these corners have been selected as the filter markers knowing their position with respect to the centre of mass. Each marker has been called with an alphabetical letter in order to have a clear identification: hence, there is Marker A, B, C, D, E, F, G, and H. The position vector of each marker, expressed in Table 5.1, is provided with respect to the centre of mass of ENVISAT; therefore there is the need to express the vector $\mathbf{v}_{CM}$ that identifies the centre of mass position with respect to the geometrical centre.

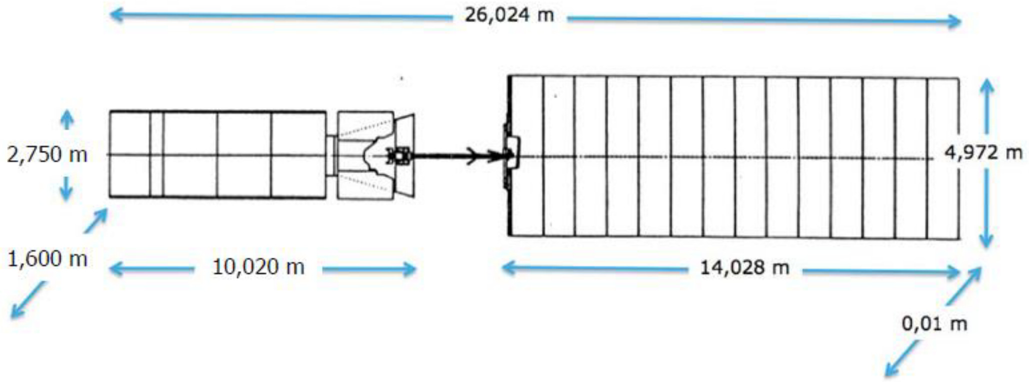

Figure 5.2: ENVISAT dimensions, image from [1]

$$\mathbf{v}_{CM} = \begin{pmatrix} -3.905 \\ -0.009 \\ +0.003 \end{pmatrix} \tag{5.22}$$

where each component is expressed in meters and the geometrical centre position is shown in Figure 5.3, along with the axis direction and the markers

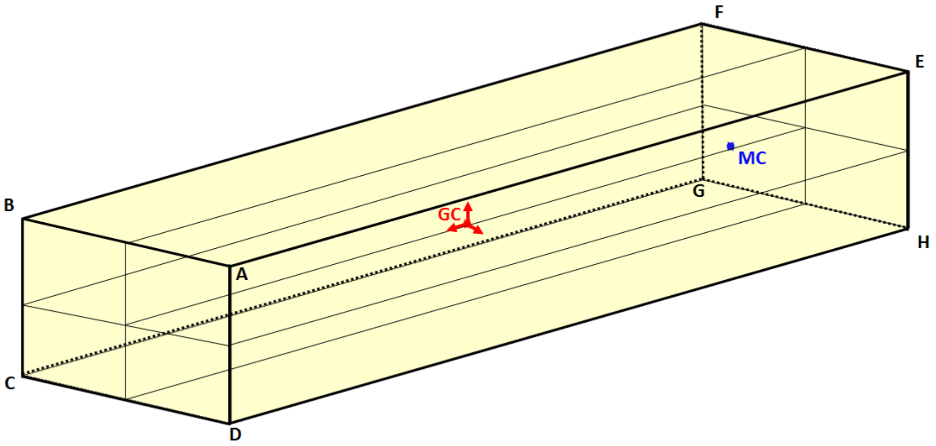

Figure 5.3: Markers, Geometrical Centre (CM), and Centre of Mass (MC) position.

position. Consequently, the position of each marker, $\mathbf{v}_i$, is calculated as simple vector summation

$$\mathbf{v}_i = \mathbf{s}_i - \mathbf{v}_{CM} \qquad for\ i = A, \ldots, H \tag{5.23}$$

where $\mathbf{s}_i$ is the marker position vector with respect to the geometric centre and $\mathbf{v}_{CM}$ is the vector from the geometrical centre to the centre of mass. Figure 5.4 gives a visual representation of the vector summation.

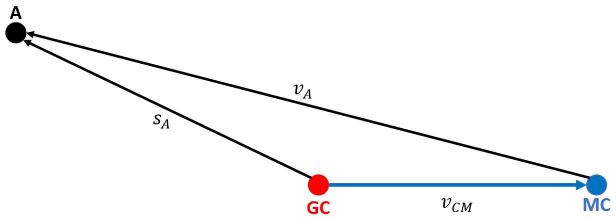

Figure 5.4: Marker A vector disposition.

The left part of Figure 5.5 illustrates on how ENVISAT appears in the chaser camera. The right part of the picture shows the system from another perspective; the red mark in the figure represents the centre of mass.

5.3.2 Measurement Equations

The state vector has 12 components that can be divided in four parts, divided into four equal parts. Each part, composed by three elements, describes one

Marker	$\mathbf{s}_i$ [m]			$\mathbf{v}_i$ [m]		
A	+5.010	+1.3750	+0.800	+8.9150	+1.3840	+1.5970
B	+5.010	+1.3750	-0.800	+8.9150	+1.3840	-1.6030
C	+5.010	-1.3750	-0.800	+8.9150	-1.3660	-1.6030
D	+5.010	-1.3750	+0.800	+8.9150	-1.3660	+1.5970
E	-5.010	+1.3750	+0.800	-1.1050	+1.3840	+1.5970
F	-5.010	+1.3750	-0.800	-1.1050	+1.3840	-1.6030
G	-5.010	-1.3750	-0.800	-1.1050	-1.3660	-1.6030
H	-5.010	-1.3750	+0.800	-1.1050	-1.3660	+1.5970

Table 5.1: ENVISAT markers position vectors with respect to its centre of mass (with no propellant)

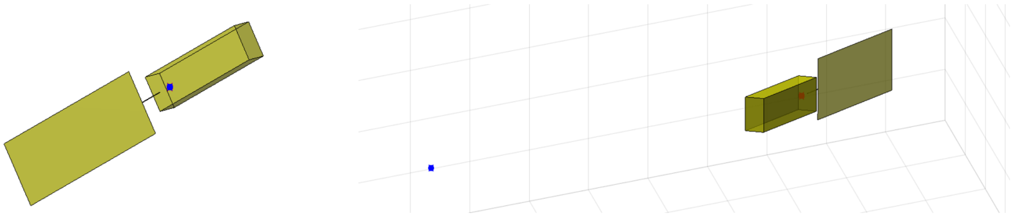

Figure 5.5: ENVISAT model as seen from the chaser (left) and external representation of the implemented simulation (right): red point is ENVISAT centre of mass, whereas blue point is the chaser centre of mass (equal to camera position)

aspect of the attitude on ENVISAT:

$$\mathbf{x} = \left(x, yz, \dot{x}, \dot{y}, \dot{z}, p_1, p_2, p_3, \omega_{r,x}, \omega_{r,y}, \omega_{r,z}\right)^T \tag{5.24}$$

Where the four parts are: the relative position between target and chaser centres of mass, relative velocity of the centre of mass, MRP, and angular velocities. Consequently, there is the need to evaluate the marker position from the knowledge of the state in order to compare, in the update part of the algorithm, the predicted measurement with the true one, which comes from the camera system.

Being $\mathbf{u}$ the position vector of the chaser centre of mass with respect to the ENVISAT centre of mass, the measurements are calculated separately for each single marker in the following way. The marker position vector $\mathbf{v}_i$ is expressed in the target reference frame; then, it is transformed in the chaser reference frame, i.e. through the multiplication of the rotation matrix $\mathbf{\Gamma}$. Then, by a simple vector difference, the position of each single marker with respect to the chaser can be evaluated:

$$\mathbf{z}_i = \mathbf{\Gamma}\mathbf{v}_i - \mathbf{u} \qquad for\ i = A, \dots, H \tag{5.25}$$

where $\mathbf{z}_i$ is the position of one marker with respect to the chaser centre of mass and the rotation matrix $\mathbf{\Gamma}$ comes from the knowledge of the MRP (which are part of the state vector) according to Equation (5.17). Figure 5.6 shows the vectors disposition in the measurement equations.

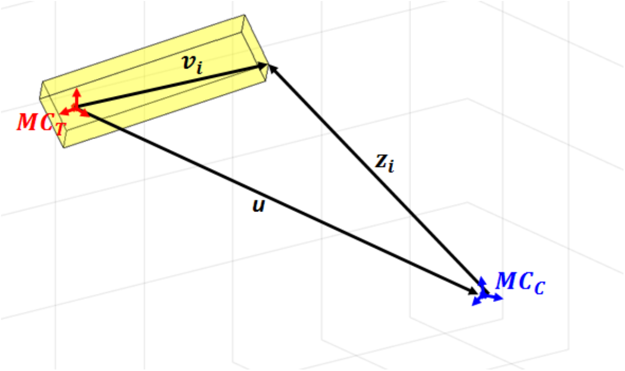

Figure 5.6: Vectors disposition for the measurement equation.

Regarding the measurement generation, the true state of ENVISAT is computed through the integration of the dynamics equations. Afterwards, noise is introduced in the measurements as an exponentially correlated random variable according the following model.

$$E(k+1) = KE(k) + \sqrt{1-K^2} \cdot \mathcal{N}(0,\sigma) \tag{5.26}$$

$$K = e^{\frac{-1}{f\tau}} \tag{5.27}$$

Where E is the error with respect to the true state, $\mathcal{N}(0,\sigma)$ is a random number generated with a Gaussian distribution (null mean and σ as standard deviation), f is the frequency in which the measurements are taken, and τ is the autocorrelation time. In this way, the error at time $k+1$ is exponentially correlated to the error at the previous step, time k, and this correlation decays with a time scaled by a time defined by τ [16].

Figure 5.7 is reported in order to show how the autocorrelated noise behaves. The figure shows the noise for one of the three components of a marker position vector, in particular along the $\hat{\mathbf{i}}$-axis direction.

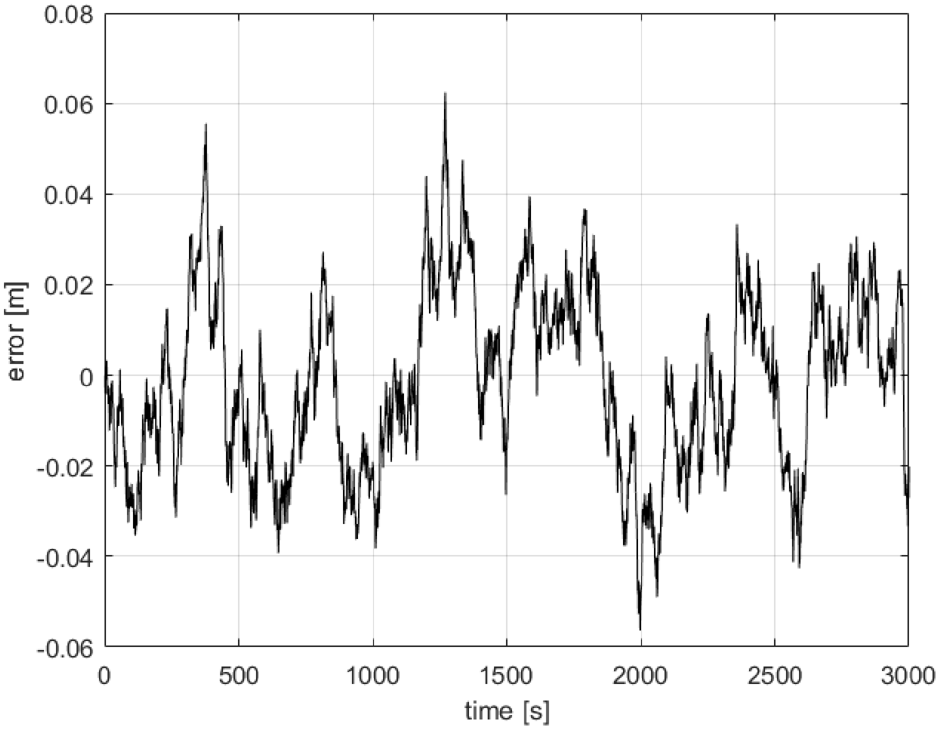

Figure 5.7: Error in the measurement of the position along the chaser LVLH $\hat{\mathbf{i}}$-axis.

5.3.3 Marker Visibility

The presented measurement model is based on the position of the 8 corners of ENVISAT main body. However, the camera is not able to locate all the markers position in one single frame due to the fact that ENVISAT's structure will cover some markers. Therefore, the filter does not work with the whole set of markers but it needs to change its measurements frame by frame depending on which marker is visible. Consequently, the measurement vector resulting from Equations (4.25) and (4.32) changes its size depending on how many markers are visible. Thus, since each marker contributes to the observation with the three components of its position, the measurement vector $\hat{\mathbf{z}}$ will have $3 \cdot i$ components, where $i = 0, \ldots, 8$.

The visibility and association of a corner to the correspondent marker is fundamental since a larger number of located markers tends to improve the estimation accuracy.

A study on the marker visibility has been performed. When thinking about visibility of the corners of a parallelepiped, it is better to focus on which face of the parallelepiped is visible and then associate the corners to each visible face. Considering ENVISAT body as free, with no other external objects that can cover it from the camera, there could be only three different options: 1, 2 or 3 faces are visible. Therefore, the set of numbers of visible markers is not a continuous set. Since each face defines 4 markers, the camera sees 4 markers if 1 face is visible, 6 markers if 2 faces are visible, and 7 markers if 3 faces are visible.

Markers visibility then relies on the correct association of ENVISAT's faces to the processed image from the camera. The process is performed by using a set of unit vectors $\hat{\mathbf{n}}_i$, with $i = \alpha, \dots, \zeta$, see Table 5.2, that defines the faces. Table 5.2 connects each face to the corresponding markers; consequently, once the visible faces are identified, the visibility of the markers is completely determined.

Face	$\hat{\mathbf{n}}_i$	**Markers Seen**
$\boldsymbol{\alpha}$	$\hat{\mathbf{n}}_\alpha = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$	A - B - C - D
$\boldsymbol{\beta}$	$\hat{\mathbf{n}}_\beta = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$	A - B - E - F
$\boldsymbol{\gamma}$	$\hat{\mathbf{n}}_\gamma = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$	A - D - E - H
$\boldsymbol{\delta}$	$\hat{\mathbf{n}}_\delta = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$	E - F - G - H
$\boldsymbol{\epsilon}$	$\hat{\mathbf{n}}_\epsilon = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$	C - D - G - H
$\boldsymbol{\zeta}$	$\hat{\mathbf{n}}_\zeta = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$	B - C - F - G

Table 5.2: ENVISAT faces and visible markers relations

The requirement for the face visibility is expressed by Equation(5.28). If the scalar product between the relative chaser-target position vector and the unit vector perpendicular to the face is negative, it means that the face is looking forward the camera and the markers associated to the face are visible.

$$\mathbf{u} \cdot \hat{\mathbf{n}}_i < 0 \qquad for\ i = \alpha, \dots, \zeta \tag{5.28}$$

The marker visibility calculation is been encoded inside the filter itself. However, this information can come from an external software, i.e. image processing, as additional input for the filters. Both cases have been implemented in this thesis, but, since the computational time of the markers visibility is of interest, only the case in which the filters calculate visibility by themselves has been reported in the results. A detailed architecture of the software is presented in the following section.

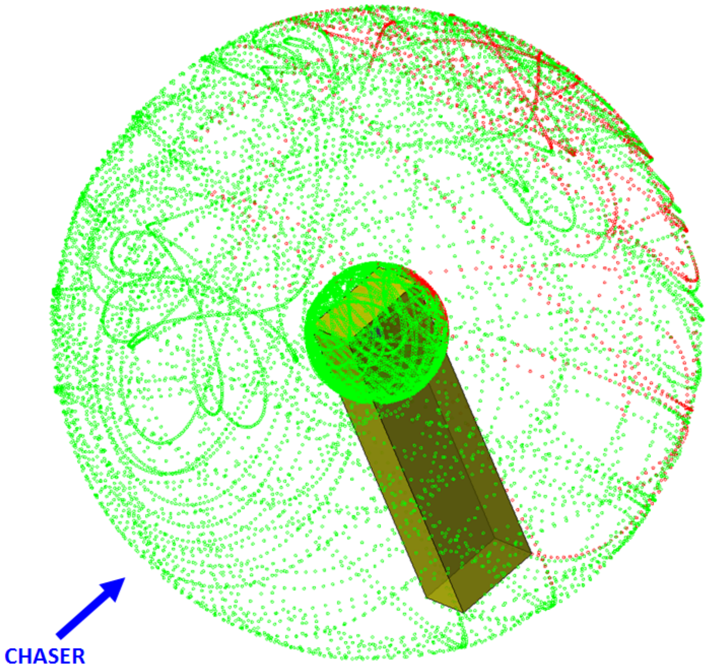

Figure 5.8: Set of visible and hidden markers in simulation of 3000 seconds with a measurement frequency of 1Hz. Solar panel omitted.

The filter uses the states at the beginning of each observation to predict the marker that will be visible at the next step and thus it prepares itself to receive the correct amount of measurements from the camera. It has hard encoded the ENVISAT faces disposition, according to versors $\hat{\mathbf{n}}_i$, and the face-markers relation of Table 5.2: therefore it predicts the marker visibility thanks to condition (5.28).

The filters handle markers visibility in a binary mode, assigning each marker value 1 if it is visible or 0 if it is not. It is possible to give a visual idea of the markers position through time by applying the dynamics presented previously to the simulation of Figure 5.5.

The results of a simulation are reported in Figure 5.8 to show the results of the marker visibility analysis during the ENVISAT motion. It can be noted that, in time, the markers move on two spheres and, more importantly, that the position of the hidden markers is concentrated in the region opposite to the chaser.

Figure 5.9 points out how the hidden markers are located all in the same semi-spheres, which are opposite to the chaser position.

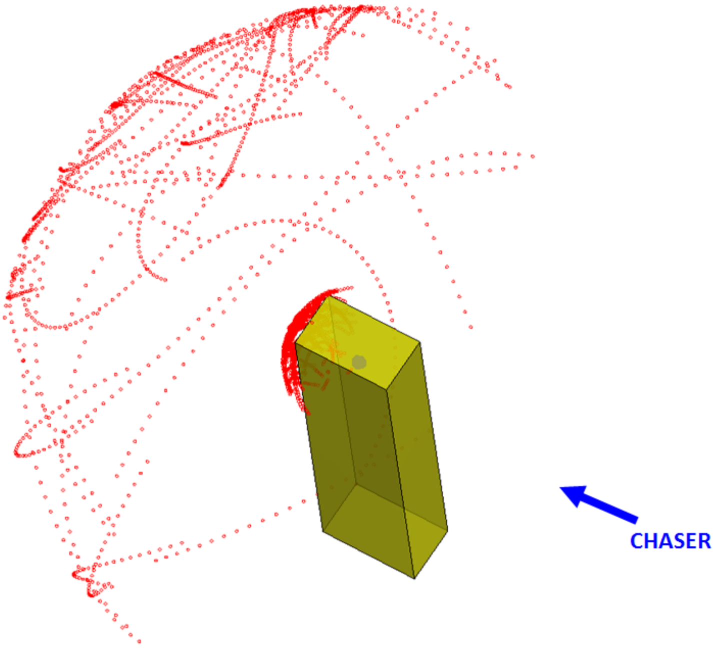

Figure 5.9: Set of hidden markers in a simulation of 3000 seconds with a measurement frequency of 1Hz. Solar panel omitted.

5.4 Software architecture

The software architecture is made up by different sections and it is shown in Figure 5.10.

Offline The measurements are generated offline and the dynamics of the system are integrated using a Runge-Kutta 78 scheme once a set of initial condition is provided. The position of the markers are computed and noise is

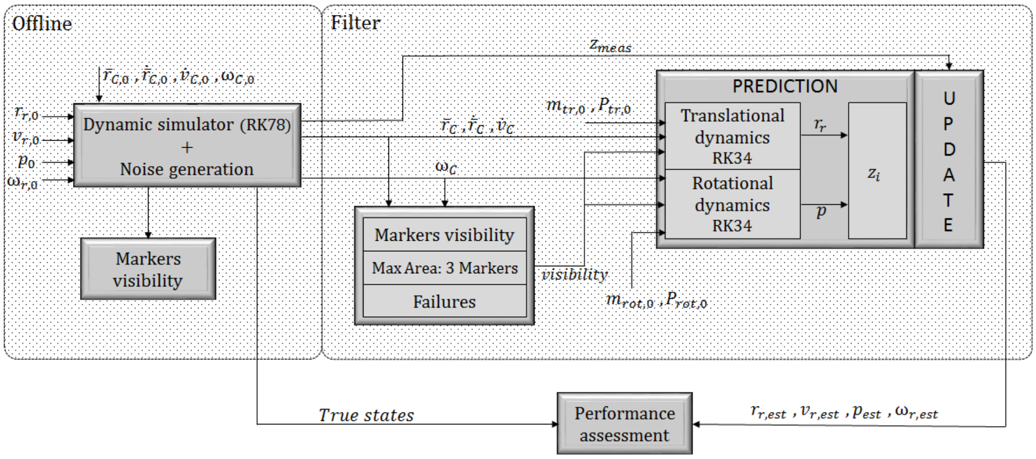

Figure 5.10: Software Architecture

added with the exponentially correlated random model described previously. The measurements creation works with the knowledge of the propagated dynamics, therefore the same precision gained through the Runge-Kutta78 integrator is reached. All these computations are performed in advance since they are common to each different kind of filter; therefore the outputs are loaded in a memory buffer before running the filters.

A file with the information of the marker visibility is here created. Depending on which type of filter is used, this file can be loaded in the filter or omitted, leading the filter itself to deal with analysis of the marker visibility.

Filtering For the filtering, the measurement model couples the translational and rotational information. However, the dynamics propagation can be decoupled into its translational and rotational part, leading to a faster and more efficient estimation of the relative translational states (relative position $\mathbf{r_r}$ and relative velocities $\mathbf{v_r}$) and of the relative rotational states (MRPs $\mathbf{p}$ and angular velocities $\omega_{\mathbf{r}}$). In this way, even if the state vector needs to be 12 components long, (which means 12 DA variables have to be initialized in the EKFDA), the relative dynamics propagation can split the vector into two separate parts of 6 components and propagate the translational and rotational model in parallel. The propagation inside the filters uses a different kind of integrator with respect to the one in the “dynamics simulator” part: a 4^{th}-order Runge-Kutta integrator is exploited. In all the filters the required markers positions and chaser absolute states are loaded at the beginning and an initial estimate of the relative states, in terms of mean and covariance, has to be provided. Moreover, before stating the estimation, the filter uses the information of the previous step to calculate the markers visibility, if it is not given. Depending on which simulation it is demanded, the filter can work with the whole set of markers or can limit the measurements to three markers (markers selection will be later explained). Moreover, measurement failures can be added to the simulation. Finally, the estimated relative states are compared with the true states propagated by the dynamics simulator to assess the performance of the filters.

5.5 Simulation Results

Referring to Figure 5.5, it can be seen how the chaser and the target are assumed to lie on the same orbit at a reasonable distance. The numerical analysis is thus performed once a set of initial condition for the relative pose of the chaser with respect to the target is selected: Table 5.3 reports the state values used as starting point for the sequential estimation process. The nominal attitude is initialized randomly, while the nominal angular velocity is selected in order to have an absolute value of about 2.5 deg/s.

Tr. Dyn.		Rot. Dyn	
x(m)	-0.002	p_1	-0.367
y(m)	-31.17	p_2	-0.590
z(m)	0	p_3	-0.570
$\dot{x}$(m/s)	-3.5e-6	$\omega_{r,x}$(rad/s)	0.02
$\dot{y}$(m/s)	-2.0e-6	$\omega_{r,y}$(rad/s)	0.02
$\dot{z}$(m/s)	0	$\omega_{r,z}$(rad/s)	0.04

Table 5.3: Initial condition

5.5.1 Robustness and Accuracy

This first part of the results aims to show the sensitivity of the filter to acquisition frequency and initial uncertainty. Therefore, the filters are tested using a discrete set of frequencies and with different initial standard deviations. For each case, a Monte Carlo-based sensitivity analysis is performed to obtain a statistical assessment of the performance of the filter. The analysed cases are reported in Table 5.4 and Table 5.5, where $\sigma_{i,0}$ indicates the initial standard deviation for the variable i and $\mathbf{K}$ is the correspondent amplification factor. As a consequence, a total of 15 different scenarios have been performed, by combining frequency and amplification factor values.

Tr. Dyn.		Rot. Dyn	
$\sigma_{r_r,0}$(m)	$\mathbf{K}*1$	$\sigma_{p_r,0}$	$\mathbf{K}*0.002$
$\sigma_{v_r,0}$(m/s)	$\mathbf{K}*0.1$	$\sigma_{\omega_r,0}$(rad/s)	$\mathbf{K}*0.1$

Table 5.4: Initial Standard Deviations

$\mathbf{K}$		1	5	10	
Freq. (Hz)	0.05	0.1	0.5	1	3

Table 5.5: Amplification Factor **K** and Frequencies values

Before explaining how each case has been analysed, it is important to point out how the selection of the sensors standard deviations σ_i^s has been made. These values are the ones that characterize matrix $\mathbf{R}$ in the filter prediction part, especially in Equation(4.26) for the EKFDA, and Equation (4.33) for the UKF and UKFDA. The sensors standard deviations are usually quite different (one order of magnitude) when referring to the different dynamics of the observed object, i.e. translational and rotational dynamics [16]. However, the simulation is based on the evaluation of the marker position, which can be considered as a series of translational only information of many separate points. Therefore, the decision of relying on the sensor standard deviation associated to a common camera has been made. The resulting values are expressed in Table 5.6.

Sensors Std. Dev.	
σ_x^s	0.02
σ_y^s	0.02
σ_z^s	0.03

Table 5.6: Standard deviation for the measurement noise.

As stated above, a Monte Carlo sensitivity analysis is performed for each of the 15 different cases. For each case, a set of 1000 samples is generated around the true initial conditions, according to the statistics. However, not the whole set has been used for the analysis, but only the furthest 100 samples are selected and used as initial estimate for the relative states in the filter. In this way, a more reasonable time-consuming analysis has been achieved where only the worst case conditions, in which the non-linearities are expected to play a prominent role, are deeply studied.

Afterwards, the performance are assessed by analysing the root mean square error (RMSE), in the following statistical way.

$$_{filtern}\bar{\mu} = \frac{\sum_{i=1}^{100} RMSE_i}{100} \tag{5.29}$$

$$_{filtern}\sigma_{\bar{\mu}} = \sqrt{\frac{\sum_{i=1}^{100} (\bar{\mu} - RMSE_i)^2}{100}} \tag{5.30}$$

where in $RMSE_i$ the subscript indicates the i^{th} simulation while $_{filtern}\bar{\mu}$

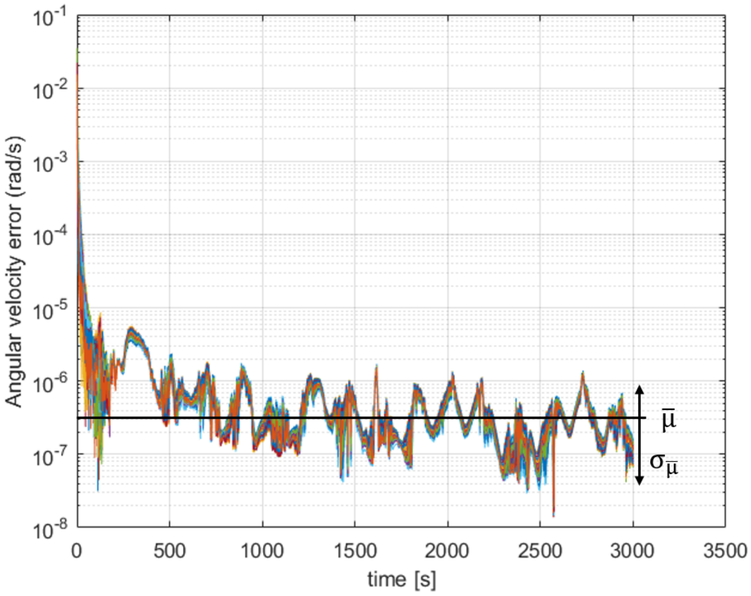

Figure 5.11: Graphical representation of statistical indexes. Angular velocity error evaluation. UKF filter for a 3000 seconds simulation with a frequency of 1Hz and amplification factor $\mathbf{K} = 1$.

and $_{filtern}\sigma_{\bar{\mu}}$ are, respectively, the mean and the relative standard deviation of the RMSE, considering the filter expressed by the subscript $filtern$. This subscript expresses the associated filter of the simulation and it is divided into two parts: the $filter$ represents the type of filter, thus it could stand for UKF, UKFDA or EKFDA; the n indicates the filter order, hence it is a number. Figure 5.11 gives a visual representation of the quantities expressed by Equations (5.29) and (5.30): $_{filtern}\bar{\mu}$ gives the accuracy level of the filter, while $_{filtern}\sigma_{\bar{\mu}}$ informs about the dispersion around the mean.

According to this sensitivity analysis, convergence is considered reached only if the final error at the end of the simulation has a mean at least one order of magnitude smaller with respect to the initial error value dictated by the initial condition sample. Only the samples that achieved convergence were considered in Equations (5.29) and (5.30).

Figure 5.12 shows the software architecture for the nominal condition associated to the results reported in the following subsection. The nominal condition uses the whole set of markers without the introduction of any measurement failure. Other settings will be later analysed.

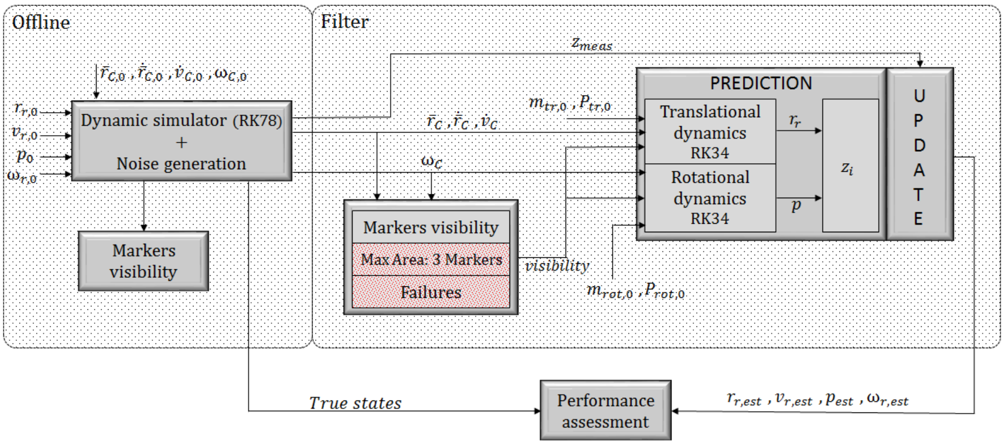

Figure 5.12: Software architecture. Nominal Condition.

5.5.2 Results

The tables with the performance of the filters are reported in this subsection. However, it is helpful to show, in a graphical way, how the values reported in the tables directly connect to the performance on the the translational and the rotational dynamics.

Figure 5.13 shows the translational part of the filter results, using a UKF simulation as example. The translational dynamics is almost linear since the chaser is close to the target and on the same orbit, which is nearly circular. As a consequence, high-order filters and UT-based ones are expected not to provide significantly better performance with respect to the classical EKF. On the other side, the rotational dynamics shows higher nonlinearities, thus the RMSE standard deviation is expected to be larger with respect to the simple translational one: Figure 5.14 shows this behaviour.

The table results are evaluated only considering the sample values that lead to convergence. If a failure occurred, or convergence was not reached, the sample was not considered in the RMSE statistics and therefore the summation in Equations (5.29) and (5.30) is performed with fewer information. Tables 5.7 and 5.8 show how many samples, out of 100, achieved valid results.

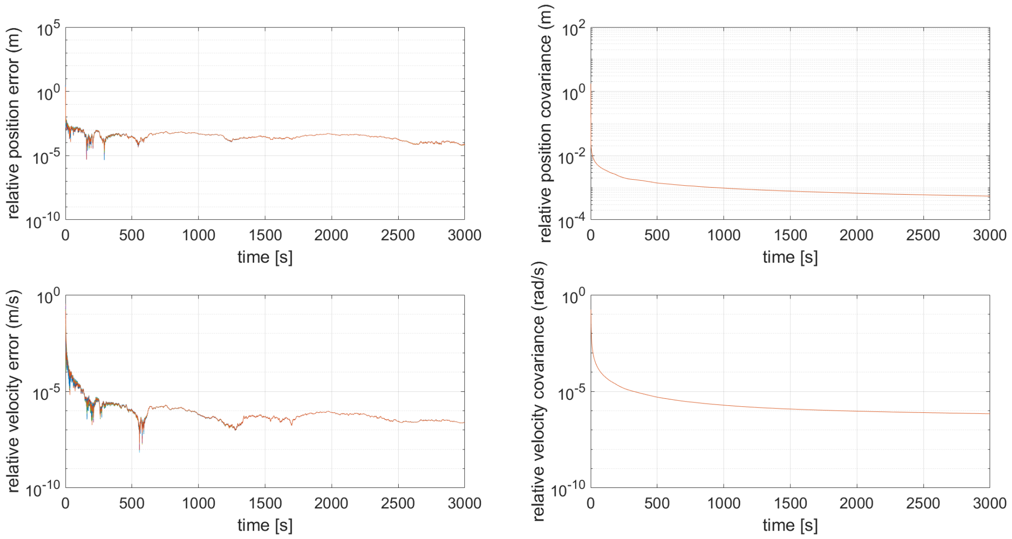

Figure 5.13: UKF. Translational results. 3000 seconds simulation with a frequency of 1Hz and amplification factor $\mathbf{K} = 1$. North-West: relative position error. North-East: relative position covariance. South-West: relative velocity error. South-East: relative velocity covariance.

It can be seen that the rotational dynamics has lower convergence ratio with respect to the translational one, especially at low frequencies.

The results in the tables show how each filter accuracy improves with higher frequency. However, the filters are able to achieve solid convergence even with low frequency, having some problems only when the amplification factor is 10, which means 100 times larger covariance with respect to the nominal case. Therefore, the implemented filters are robust and accurate both for the translational and rotational pose estimation of the target. Almost in each step, the filter corrects its prediction values thanks to the information from 7 different markers, since it is rare to have only two faces of the target on sight, and almost impossible to have only one. However, some differences between the filters must be underlined. There is no significant improvement in the performance of the filter by passing from the EKFDA1 to the UKFDA, UKF, and EKFDA2 in the translational part: the reason is due to the fact that translational dynamics is almost linear, cancelling the advantages given by nonlinear filters. Only Table 5.9 and Table 5.10, which refer to the UKFDA2 filter, are reported for the translational dynamic RMSE mean and standard deviation since each filter leads to the same error level in the estimation process: in the Appendix all the tables referring to all the other filters are reported to check the results of each single simulation.

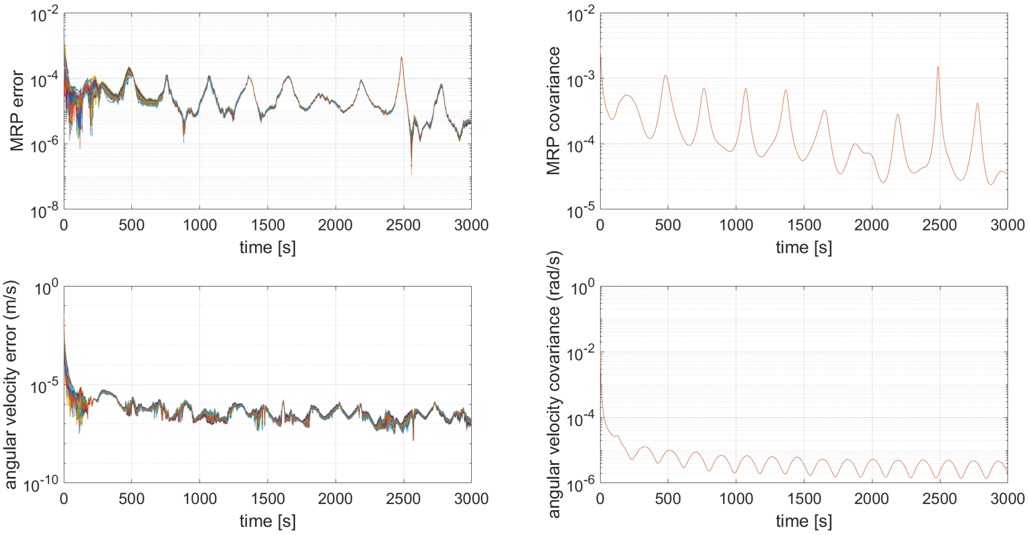

Figure 5.14: UKF. Rotational results. 3000 seconds simulation with a frequency of 1Hz and amplification factor **K** = 1. North-West: MRP error. North-East: MRP covariance. South-West: angular velocity error. South-East: angular velocity covariance.

EKFDA1				EKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	100	63	49	**0.05**	99	56	49
0.1	100	100	81	**0.1**	100	100	90
0.5	100	100	100	**0.5**	100	100	100
1	100	100	100	**1**	100	100	100
3	100	100	100	**3**	100	100	100
UKF				UKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	100	70	60	**0.05**	100	69	58
0.1	100	100	83	**0.1**	100	100	83
0.5	100	100	100	**0.5**	100	100	100
1	100	100	100	**1**	100	100	100
3	100	100	100	**3**	100	100	100

Table 5.7: Success rate for each combination of frequency and amplitude factor. Translational dynamics

EKFDA1				EKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	8	6	2	**0.05**	8	17	3
0.1	59	41	18	**0.1**	69	79	36
0.5	100	100	100	**0.5**	100	100	100
1	100	100	100	**1**	100	100	100
3	100	100	100	**3**	100	100	100
UKF				UKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	12	11	3	**0.05**	12	11	3
0.1	64	57	24	**0.1**	64	57	24
0.5	100	100	100	**0.5**	100	100	100
1	100	100	100	**1**	100	100	100
3	100	100	100	**3**	100	100	100

Table 5.8: Success rate for each combination of frequency and amplitude factor. Rotational dynamics

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	3.93e-3	1.74e-1	6.89e-1	**0.05**	5.51e-6	3.10e-4	1.24e-3
0.1	1.42e-3	1.10e-2	2.28e-1	**0.1**	2.63e-6	1.64e-5	3.63e-4
0.5	6.33e-4	6.29e-4	7.86e-4	**0.5**	9.34e-7	9.60e-7	1.34e-6
1	2.88e-4	2.81e-4	2.77e-4	**1**	5.08e-7	4.90e-7	4.68e-7
3	8.82e-5	8.96e-5	9.50e-5	**3**	1.52e-7	1.55e-7	1.65e-7

Table 5.9: $_{UKFDA2}\bar{\mu}$ sensitivity analysis for translational performance.

On the other side, regarding the rotational part of the simulation, higher-order EKF and UKF (and UKFDA) provide more accurate state estimations. Improvements are evident especially at low frequencies and in cases with high amplification factor **K**. EKFDA1 has larger RMSE standard deviation with respect to the other filters. Moreover, Table 5.7 and Table 5.8 show that EKFDA1 is the filter with the lowest success rate, thus the first 2 rows of all the tables related to EKFDA1, and the first row of the other tables must be read having in mind that only few of the samples were able to reach convergence. However, these two tables are filled with the samples that were able to achieve a mean error, in the last part of the simulation, one order

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.05e-3	3.27e-1	5.11e-1	**0.05**	1.89e-6	6.33e-4	1.02e-3
0.1	6.11e-5	2.60e-2	4.07e-1	**0.1**	1.24e-7	3.66e-5	7.15e-4
0.5	2.53e-6	4.90e-5	2.94e-4	**0.5**	2.68e-9	6.39e-8	5.53e-7
1	4.53e-7	7.07e-5	2.98e-5	**1**	6.51e-10	1.18e-8	4.39e-8
3	8.37e-8	1.36e-6	6.52e-6	**3**	1.35e-10	2.25e-9	1.07e-8

Table 5.10: $_{UKFDA2}\sigma_{\bar{\mu}}$ sensitivity analysis for translational performance.

of magnitude smaller with respect to the initial one. This constrain left out many samples that were able to achieve convergence but in a softer way: Figure 5.15 is such a good example to show how the EKFDA2 converges with almost all of the samples even if the related value in Table 5.8, in row 0.05 Hz and column **K** = 1, indicates that only 8 samples converge. If the frequency is reduced, the reported accuracy of the filters will stabilize on the convergence level (since the samples with mean error bigger than the threshold will be left out) until no sample would reach convergence.

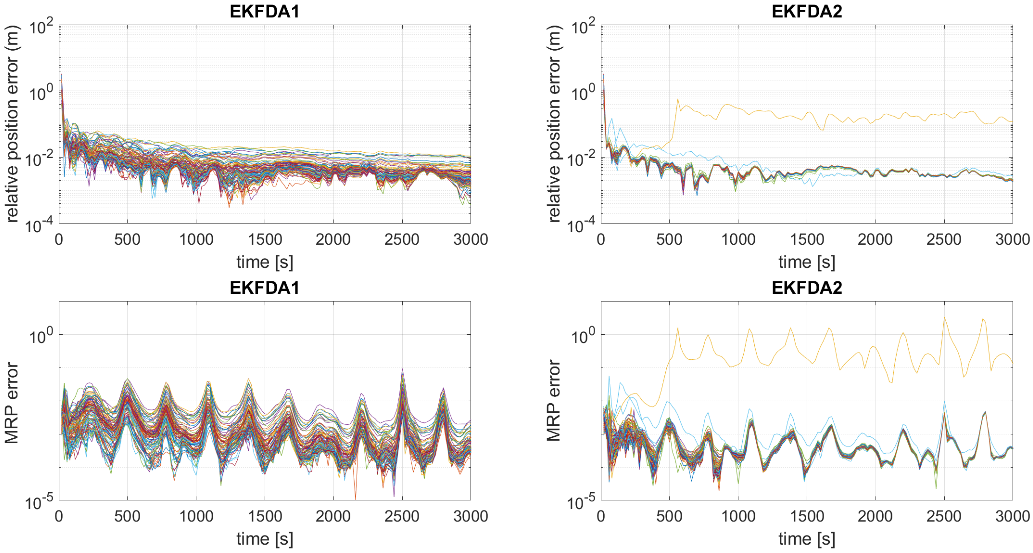

Figure 5.15: EKFDA1 and EKFDA2 accuracy for the whole set of 100 samples. Simulation of the case with frequency 0.05 Hz and **K** = 1. Comparison for the relative position estimation, upper row, and for the Modified Rodriguez Parameters estimation, lower row.

Filters that work with the full knowledge of the first two moments work better with respect to the filter based on linearization. However there are differences among them. EKFDA2 seems to appear the filter with the best accuracy, especially for low frequencies and high uncertainties. The filters based on the UT, UKF and UKFDA, produce the same results (with few exceptions, for low frequencies with high uncertainties): as already stated, their difference is not on the accuracy level but on the computational time requested by the filter itself, as will be later shown. The introduction of the DA inside the UKF has the purpose to achieve a faster filter, without affecting accuracy.

A visual representation of the improvements of EKFDA2 with respect to EKFDA1 is shown in Figure 5.15 where the case with frequency of 0.05 Hz and amplification factor $\mathbf{K} = 1$ is considered. The figure shows the error of both relative position and MRP for all the 100 samples: in EKFDA1 the high standard deviation of the RMSE can be seen on how the lines related to each sample toil to merge together; while the EKFDA2 shows a more narrow behaviour toward convergence. Thus, high order filters have improved the estimation process more on dispersion, $_{filtern}\sigma_{\bar{\mu}}$, than on mean level of accuracy, $_{filtern}\bar{\mu}$.

The accuracy results connected to Figure 5.15 are reported in Table 5.11 to Table 5.14, referring to the rotational performance. By directly comparing Table 5.12 and Table 5.14 it can be seen how $_{EKFDA1}\sigma_{\bar{\mu}} >_{EKFDA2} \sigma_{\bar{\mu}}$ and the few exceptions (at low frequency) are due to the low number of samples with convergence in Equations (5.29) and (5.30).

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	8.90e-4	1.98e-3	5.13e-3	**0.05**	1.82e-5	3.48e-5	4.31e-5
0.1	4.43e-4	1.33e-3	2.64e-3	**0.1**	9.46e-6	2.51e-5	3.24e-5
0.5	9.46e-5	1.06e-4	2.27e-4	**0.5**	1.93e-6	2.31e-6	5.37e-6
1	5.16e-5	5.39e-5	6.87e-5	**1**	3.93e-7	5.30e-7	1.11e-6
3	9.44e-6	1.06e-5	1.44e-5	**3**	1.52e-7	1.95e-7	3.55e-7

Table 5.11: $_{EKFDA1}\bar{\mu}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.49e-4	6.31e-4	3.34e-4	**0.05**	3.84e-4	1.94e-5	3.79e-5
0.1	6.96e-5	1.17e-4	2.20e-3	**0.1**	2.75e-6	2.40e-5	2.46e-5
0.5	1.81e-6	3.27e-5	2.13e-4	**0.5**	8.81e-8	1.01e-6	4.48e-6
1	1.24e-6	8.82e-6	3.27e-5	**1**	1.21e-8	2.07e-7	9.52e-7
3	1.37e-7	1.44e-6	7.30e-6	**3**	6.16e-9	6.63e-8	2.79e-7

Table 5.12: ${}_{EKFDA1}\sigma_{\bar{\mu}}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.36e-4	1.21e-3	2.82e-3	**0.05**	2.35e-5	2.51e-5	2.17e-5
0.1	4.49e-4	7.60e-4	1.23e-3	**0.1**	1.05e-5	1.64e-5	2.24e-5
0.5	9.46e-5	1.04e-4	2.13e-4	**0.5**	1.96e-6	2.25e-6	5.01e-6
1	5.07e-5	5.30e-5	6.79e-5	**1**	3.88e-7	5.24e-7	1.10e-6
3	9.17e-7	9.98e-6	1.41e-5	**3**	1.49e-7	1.91e-7	3.50e-7

Table 5.13: ${}_{EKFDA2}\bar{\mu}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.54e-5	4.01e-4	2.21e-3	**0.05**	6.56e-7	6.79e-6	8.79e-6
0.1	9.47e-6	6.23e-4	1.10e-3	**0.1**	4.29e-6	1.19e-5	2.38e-5
0.5	1.75e-6	3.05e-5	1.96e-4	**0.5**	8.47e-8	9.51e-7	4.17e-6
1	1.23e-6	8.67e-6	3.24e-5	**1**	1.03e-8	2.02e-7	9.41e-7
3	1.23e-7	1.38e-6	7.25e-6	**3**	5.58e-9	6.45e-8	2.77e-7

Table 5.14: ${}_{EKFDA2}\sigma_{\bar{\mu}}$ sensitivity analysis for rotational performance.

5.5.3 Computational Time

The other performance analysis concerns the computational time. Table 5.15 shows the software computational time performed on a 3000 seconds simulation. The characteristic time used to describe the performance of the filters at each frequency has been evaluated as a mean among all the simulations connected to the samples that lead to convergence. Therefore, any failure has been left out from the computational time analysis. The computational time mean is evaluated in the following way:

$$\Psi_f = \frac{\sum_{j=1}^{3} \frac{\sum_{i=1}^{\kappa_{f,j}} \tau_{f,j,i}}{\kappa_{f,j}}}{3} \tag{5.31}$$

where Ψ_f is the computational mean time associated to frequency f ($f = 0.05, 0.1, 0.5, 1, 3$); j indicates the amplification factor $\mathbf{K}$ of the simulation ($j = 1 \rightarrow \mathbf{K} = 1, j = 2 \rightarrow \mathbf{K} = 5, j = 3 \rightarrow \mathbf{K} = 10$); $\kappa_{f,j}$ is the number of samples that achieved convergence for the simulation with frequency f and amplification factor j; and $\tau_{i,j,k}$ is the computational time of the i^{th} sample in simulation (f, j). The characteristic time Ψ_f is the mean among all the different values of $\mathbf{K}$, since it is only the acquisition frequency of the measurement the relevant figure of merit when dealing with time.

Table 5.15 purpose is to provide a comparison between the proposed filters and the data is visualized in Figure 5.16.

Freq.	EKFDA1	EKFDA2	UKF	UKFDA2
(Hz)	Ψ_f (s)	Ψ_f (s)	Ψ_f (s)	Ψ_f (s)
0.05	1.47	5.84	8.06	2.41
0.1	1.67	6.70	8.50	2.69
0.5	3.23	13.88	12.47	5.66
1	5.18	22.43	17.23	9.31
3	11.54	50.13	29.16	21.94

Table 5.15: Computational time analysis of the filters

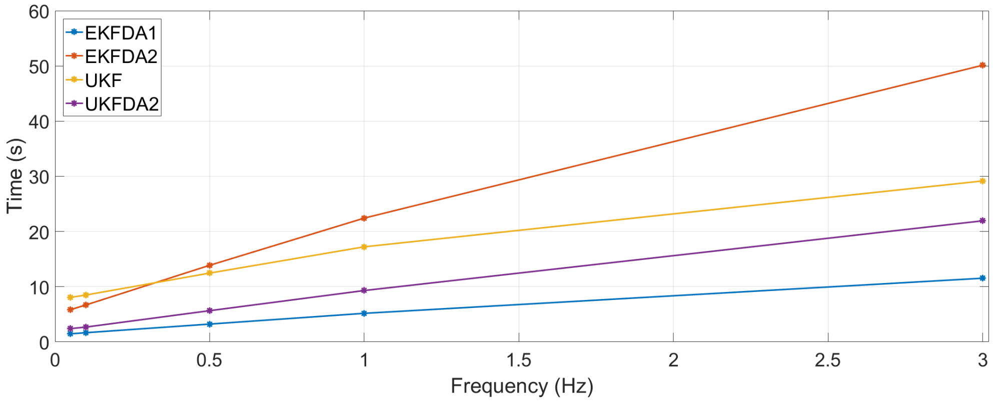

Figure 5.16: Mean computational time of the filters at different frequencies.

The linear filter EKFDA1, as expected, is the fastest one. The UKFDA has nearly the same trend of the UKF but it requires always less time with respect to the UKF. Therefore, the improvement of including DA in the UT is evident: the two filters have the same accuracy, but the one based on DA is faster in the whole frequency range. The EKFDA2 is the filter with the most demanding computational time. However, this happens only at high frequencies because, thanks to the different slope of the curve, it is able to show faster behaviour at low frequencies with respect to UKF.

5.6 Performance with three markers

The filters have been proven to work with the whole set of available markers, which means a shape-shifting measurement vector that adapts to the number of visible markers for each acquisition. A new scenario is now analysed, which relies on the assumption of the identification of a fixed number of markers: three markers. Therefore, each acquisition, a set of 3 markers, taken by the full group of visible ones, will be used to create the measurement vector. This study is also justified by the fact that the solar panel of ENVISAT may cover some of the markers and therefore the available measurements decrease due to the restricted visibility. The selection of the makers is based on a simple criterion: the filter will have to work with the 3 markers that create the triangle with the largest area on the plane of sight. The plane of view (or sight), from now on called π_1, is the ideal plane defined by the relative position chaser-target vector and passing through the target centre of mass. This plane gives information on how the camera sees the target, therefore each marker is projected on π_1 before evaluating the triangle with the largest area. Each marker projection lies on plane π_1 and is the vertex of many triangles whose number depends on how many markers are visible. In this way, the perceived area of all the triangles on plane π_1 is evaluated and maximized.

As described previously, the available number of markers, from now on referred as n, is 8. These n markers are always selected in group of 3; let k be the number of group elements. Since the order in which the markers are selected is not important, which means that group ADH gives the same information of DHA, the number of possible different groups is perfectly described by the binomial coefficient expressed in Equation (5.32) [17].

$$\Xi = \binom{n}{k} = \frac{n!}{k!(n-k)!} = 56 \tag{5.32}$$

Ξ indicates the number of all possible combinations, but not every combination is suitable for the estimation process. Furthermore, some combinations are physically impossible due to ENVISAT geometry model. Having in mind the geometrical model of ENVISAT, a parallelepiped, opposite markers are the ones that are connected by a central symmetry with centre the geometrical centre of the model. Thus, there are 4 pairs of markers that will never be part of the same group because if the first is visible, the other one will not. Looking at Figure 5.3, the four pairs are AG, BH, CE and DF, and they lead to a total of 24 *impossible* combinations.

As a result, the number of physical combinations can be found by a simple difference between Ξ and the unphysical combinations. However, not all the 56 - 24 = 32 combinations are valid for the estimation process. There is a further distinction of markers based on their relative position in the ENVISAT geometrical model. A valid combination is selected not have the 3 markers belonging to the same side of the model. Recalling Figure 5.8, this statement states that in a combination of 3 markers, they cannot all belong to the same sphere, but if two markers are in the outer sphere, then the third must be in the inner one. Therefore, a new division of groups has been performed, dividing the 32 physical combinations in *allowed* and *forbidden*. Table 5.16 shows all combinations divided by their classification. The number of the forbidden combination can be found by the binomial coefficient of 4 available markers per side grouped by 3; being two sides, the forbidden combination are $2 \times \binom{4}{3} = 8$. This leaves a total of 24 allowed combinations.

	Groups							
impossible	ABG	ACG	ADG	AEG	AFG	AGH	ABH	BCH
	BDH	BEH	BFH	BGH	ACE	BCE	CDE	CEF
	CEG	CEH	ADF	BDF	CDF	DEF	DFG	DFH
forbidden	ABC	ABD	BCD	ACD	EFG	EFH	FGH	EGH
allowed	ABE	ABF	BCF	BCG	CDH	CDG	ADE	ADH
	ACF	ACH	BDE	BDG	AEF	BEF	BFG	CFG
	CGH	DGH	AEH	DEH	BEG	DEG	AFH	CFH

Table 5.16: Groups of 3 markers divided by their nature

The filters will try to work using only the allowed combinations, but, in lack of available data, they will work with the forbidden combinations, which is way better that having a time step with no measurements.

In order to select the combination of markers that creates the largest triangle on the plane of view, every marker must be projected on π_1. The unit vector $\mathbf{u}$ perpendicular to plane π_1 (and with origin ENVISAT's centre of mass) is found by taking the relative position chaser-target vector and dividing its component by the relative module. Being $\mathbf{m}_i$, with $i = A, \ldots, H$, the vector of each marker position with respect to the target centre of mass, the projection of the marker on plane π_1 is found by subtracting to $\mathbf{m}_i$ the component directed along $\mathbf{u}$, according the following equations.

$$b_i = \mathbf{u} \cdot \mathbf{m}_i \tag{5.33}$$

$$\tilde{\mathbf{m}}_i = \mathbf{m}_i - b_i\mathbf{u} \tag{5.34}$$

Where $\tilde{\mathbf{m}}_i$ is the vector projection of the i^{th} marker lying on π_1 with origin taken at the target centre of mass, and b_i is the scalar component of the marker vector $\mathbf{m}_i$ in direction $\mathbf{u}$. Let X' be the projection on π_1 of marker X (defined by the vector $\tilde{\mathbf{m}}_X$). All the markers have been reported on the same plane: the triangle with the largest area can now be found by evaluating the areas of all the possible triangles with vertexes the visible markers by using, for example, Erone's formula.

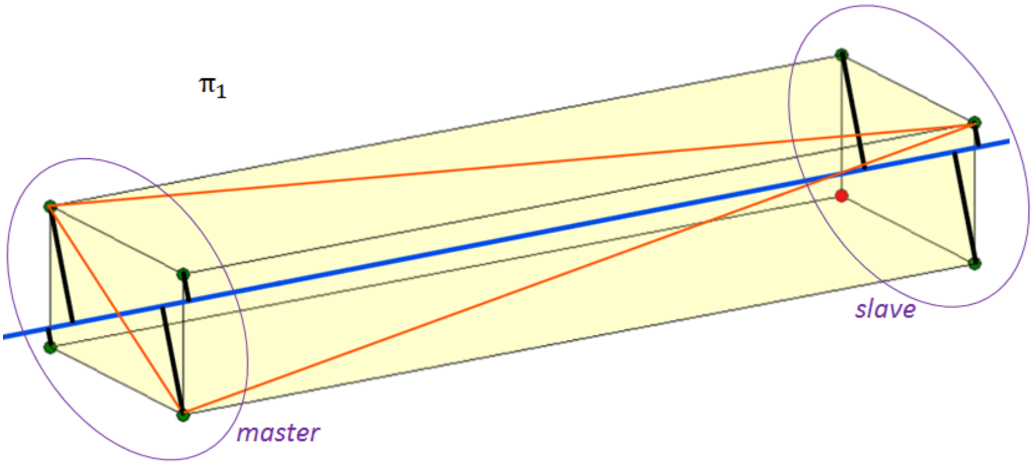

Figure 5.17: Three markers selection.

However, a more elegant, and less computational demanding, solution has been adopted. Since ENVISAT geometrical model is a slender parallelepiped with one predominant axis, axis defined by $\hat{\mathbf{n}}_\alpha$ in Table 5.2, it is convenient to think about the markers as divided into two separate groups: group SX with markers A, B, C, and D, and group DX with markers E, F, G and H. The *master* group is the one which has more visible markers, usually 4, with respect to the *slave* group, with fewer visible markers, usually 3. The projection on π_1 of the principal inertia axis defined by $\hat{\mathbf{n}}_\alpha$ can be calculated in the same way as performed in Equations (5.33) and (5.34) for the markers.

$$\tilde{\mathbf{n}}_\alpha = \hat{\mathbf{n}}_\alpha - (\mathbf{u} \cdot \hat{\mathbf{n}}_\alpha)\mathbf{u} \tag{5.35}$$

Versor $\tilde{\mathbf{n}}_\alpha$ defines a straight line on plane π_1, line r. Figure 5.17 shows plane π_1 with the visible and hidden markers, line r (in blue), *master* and *slave* group, and the triangle with the maximum area.

The markers combination that has the triangle with the largest area can be directly found by selecting the two furthest marker projections from line r in the *master* group and the nearest marker projection, again from line r, in the *slave* group. In this way, there is no need to evaluate all the possible triangle areas, but it is sufficient to compute, for each visible marker, the distance between its projection X' and line r. This distance, d_i, is evaluated by considering vector $\tilde{\mathbf{m}}_i$ as the hypotenuses of a rectangular triangle with one cathetus on line r and the other d_i itself.

$$d_i = |\tilde{\mathbf{m}}_i \wedge \tilde{\mathbf{n}}_\alpha| \tag{5.36}$$

In this way, the best *allowed* group from Table 5.16 can be rapidly found for each acquisition.

The software setting for the three markers scenario is shown in Figure 5.18 where only the introduction of measurement failures in the system is left out.

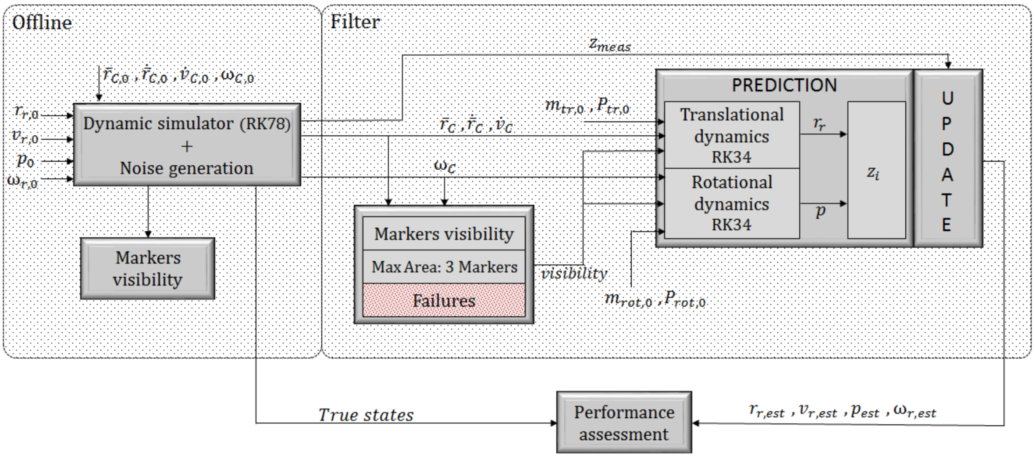

Figure 5.18: Software architecture. Three markers condition.

The results of all the simulations are reported in the appendix, where the subscript "$3M$" indicates that the simulations have been performed with the filters measurement vector limited at 3 markers. Here are reported the results form the EKFDA2 simulations are only reported.

The filter convergences in almost all the cases. A slight accuracy decrease, with respect to the filter working with all the markers, can be noted by comparing the tables reporting the RMSE mean: compare Table 5.19 with Table

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.62e-2	1.89e-1	1.11e-0	**0.05**	3.23e-5	2.98e-4	1.74e-3
0.1	1.38e-3	9.62e-3	3.86e-1	**0.1**	2.65e-6	1.48e-5	6.16e-4
0.5	6.26e-4	6.34e-4	9.37e-4	**0.5**	9.27e-7	9.90e-7	1.65e-6
1	2.94e-4	2.88e-4	3.16e-4	**1**	5.22e-7	4.92e-7	4.84e-7
3	8.87e-5	9.25e-5	1.14e-5	**3**	1.53e-7	1.59e-7	1.93e-7

Table 5.17: $_{EKFDA2}\bar{\mu}_{3M}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	6.83e-2	3.16e-1	6.07e-1	**0.05**	1.42e-4	5.19e-4	1.01e-3
0.1	2.23e-5	2.11e-2	5.98e-1	**0.1**	5.69e-8	3.22e-5	9.91e-4
0.5	3.33e-6	6.63e-5	4.59e-4	**0.5**	3.56e-9	1.04e-7	8.25e-7
1	9.55e-7	1.45e-5	9.24e-5	**1**	1.58e-9	2.26e-8	8.13e-8
3	2.15e-7	5.59e-6	3.83e-5	**3**	2.92e-10	7.60e-9	5.15e-8

Table 5.18: $_{EKFDA2}\sigma_{\bar{\mu},3M}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.12e-4	1.76e-3	-	**0.05**	2.05e-5	2.42e-5	-
0.1	4.55e-4	8.15e-4	1.71e-3	**0.1**	1.24e-5	1.80e-5	3.14e-5
0.5	8.80e-5	1.07e-4	2.95e-4	**0.5**	1.37e-6	2.16e-6	6.83e-6
1	6.46e-5	7.34e-5	1.24e-4	**1**	4.41e-7	8.38e-7	2.45e-6
3	7.76e-6	1.31e-5	3.36e-5	**3**	1.25e-7	3.04e-7	9.68e-7

Table 5.19: $_{EKFDA2}\bar{\mu}_{3M}$ sensitivity analysis for rotational performance.

5.13. However, the main difference lies on the RMSE standard deviation: which is bigger (compare Table 5.20 with Table 5.14). The 3 markers simulations spread more around the mean since the update step relies on fewer information compared to before. As a consequence, the number of failures and final uncertainty have increased. The overall performance decay is not so marked, but it is important to stress that the estimation process has become critical at very low frequencies.

Table 5.22 shows how the convergence was not achieved with the EKFDA

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	4.42e-5	3.48e-4	-	**0.05**	2.64e-6	8.49e-6	-
0.1	1.93e-5	5.00e-4	1.58e-3	**0.1**	9.24e-7	1.27e-5	3.33e-5
0.5	3.44e-6	3.87e-5	3.23e-4	**0.5**	1.57e-7	1.29e-6	7.48e-6
1	3.07e-6	2.24e-5	9.64e-5	**1**	2.49e-8	5.34e-7	2.89e-6
3	3.50e-7	7.58e-6	3.57e-5	**3**	6.71e-9	2.54e-7	1.20e-6

Table 5.20: $_{EKFDA2}\sigma_{\bar{\mu},3M}$ sensitivity analysis for rotational performance.

EKFDA1				EKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	100	57	34	**0.05**	95	52	47
0.1	100	100	73	**0.1**	100	100	87
0.5	100	100	100	**0.5**	100	100	100
1	100	100	100	**1**	100	100	100
3	100	100	100	**3**	100	100	100

UKF				UKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	100	63	53	**0.05**	100	59	48
0.1	100	100	80	**0.1**	100	100	82
0.5	100	100	100	**0.5**	100	100	100
1	100	100	100	**1**	100	100	100
3	100	100	100	**3**	100	100	100

Table 5.21: Success rate for each combination of frequency and amplitude factor. Translational dynamics. 3 Markers limitation.

in the case with 0.05 Hz and $\mathbf{K} = 10$ and fewer samples reached convergence for the other scenarios, in general. The UT-based filters have higher convergence rate of samples at low frequency with respect to DA-based EKF filters, but the EKFDA2 still shows slight better accuracy. The EKFDA1 is the least accurate filter: the difference among the others is evident, anyhow, only for the rotational part of the simulation, since the translational dynamics is quite linear. However, even if the improvement gained by UT-based and DA-based filters is not so relevant in the mean error accuracy ($_{filtern}\bar{\mu}_{3M}$), $_{EKFDA1}\sigma_{\bar{\mu},3M}$ differs from the other filters showing bigger values. This means that, espe-

EKFDA1				EKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	8	5	0	**0.05**	8	7	0
0.1	58	30	11	**0.1**	63	72	27
0.5	100	100	100	**0.5**	100	100	100
1	100	100	100	**1**	100	100	100
3	100	100	100	**3**	100	100	100
UKF				UKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	11	8	2	**0.05**	11	8	2
0.1	57	41	16	**0.1**	57	41	16
0.5	100	100	100	**0.5**	100	100	100
1	100	100	100	**1**	100	100	100
3	100	100	100	**3**	100	100	100

Table 5.22: Success rate for each combination of frequency and amplitude factor. Rotational dynamics. 3 Markers limitation.

cially in low frequency cases, the EKFDA1 samples have produced simulations that spread more around the mean value, thus proving that EKFDA2 and UT-based filters are more reliable. UKF and UKFDA show again the same results.

Figure 5.19 shows how EKFDA2 has better performance with respect to the unscented filter based on DA. The figure reports the 100 samples simulation at frequency 0.1 Hz and $\mathbf{K} = 1$: the set of EKFDA2 error lines is narrower with respect to the UKFDA2 error lines (left column).

Table 5.23 reports the computational time of the filters for the 3 markers simulations. As expected, the overall simulation time has decreased for each filter for all the frequencies. The EKFDA1 is again the fastest filter, followed by UKFDA2 with almost double computational time. The main difference between the default simulation lies on the comparison between EKFDA2 and UKF. EKFDA2 is still the slowest filter at 3 Hz. The reduced length of the measurement vector implies a faster inversion of the measurement covariance matrix, which is the most time demanding passage in the DA-based filter. Therefore, for the 3 marker simulation, the EKFDA2 reduces its computational time as the frequency decreases with a stronger slope compared to the UKF. As a result, EKFDA2 and UKF have almost the same computational time near 1 Hz and the UKF becomes the most demanding filter, in terms

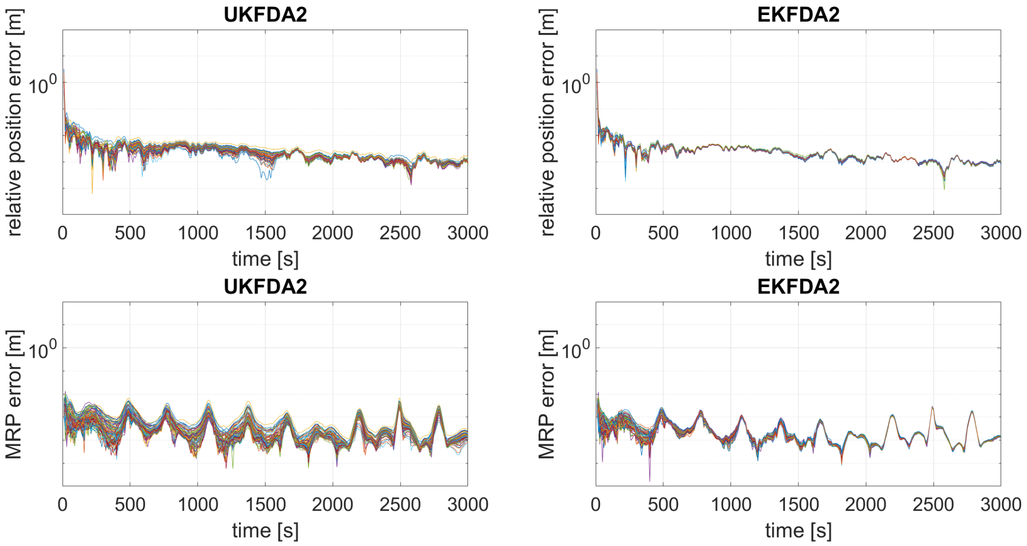

Figure 5.19: UKFDA2 and EKFDA2 accuracy for the whole set of 100 samples. 3 Markers limitation. Simulation of the case with frequency 0.1 Hz and $\mathbf{K} = 1$. Comparison for the relative position estimation, upper row, and for the Modified Rodriguez Parameters estimation, lower row.

of time, at higher frequencies.

Freq. (Hz)	EKFDA1 Ψ_f (s)	EKFDA2 Ψ_f (s)	UKF Ψ_f (s)	UKFDA2 Ψ_f (s)
0.05	1.40	5.49	7.98	2.23
0.1	1.53	6.00	8.26	2.41
0.5	2.93	10.19	11.15	4.22
1	3.37	15.25	14.29	6.43
3	6.23	30.10	20.43	13.31

Table 5.23: Computational time analysis of the filters. 3 Markers limitation.

5.7 Acquisition Failures

Measurement failures can be critical in sequential state estimation application. The lack of proper data management could lead to an unforeseen behaviour of the filters. Therefore, in this section, the filters have been tested against marker position data failure. The failure analysis has been restricted to the 3 markers limitation case, since it is the most critical scenario.

Considering the set of 3 marker, the occurrence of a failure is indicated with 1, while the correct marker is indicated with 0. Therefore, being the set of 3 elements, there are $2^3 = 8$ different combination of failures: 111 - 100 - 010 - 001 - 110 - 101 - 011 - 000. Given a failure probability level, $p = 20\%$, it is possible to calculate, using statistics, the probability of having a certain amount of failures in the set of 3 markers, shown in Table 5.24. The integer number α indicates the number of failures in a set. The probability of having α failures in a set, $\mathcal{P}_\alpha$, is evaluated according the following formula.

$$\mathcal{P}_\alpha = [p^\alpha (1-p)^{(N-\alpha)}]\gamma \tag{5.37}$$

Where N, number of elements, is 3 due to the fact that the set includes 3 markers, and γ is an integer number that tells how many combination of failures have α failures. The failures set probability $\mathcal{P}_\alpha$ shall follow Equation (5.38).

$$\sum_{\alpha=0}^{N} \mathcal{P}_\alpha = 1 \tag{5.38}$$

α	γ	$\mathcal{P}_\alpha$
3	1	0.80%
2	3	9.60%
1	3	38.40%
0	1	51.20%

Table 5.24: Probability of failures in a set of 3 markers.

Table 5.24 shows how only half of the time steps work without any failure (with probability threshold p). The simulations at high frequencies, 3 Hz and 1 Hz, are not reported since their performance resemble the simulations done without failures with a frequency of half value. The failure simulations have their own subscript, which is "$3Mfail$", and the results can be read in the tables in the appendix. The software settings for the failure condition is shown in Figure 5.10 in the software architecture section.

As expected, failures affects negatively the filters performances. However, the filters are able to reach convergence and the state estimation of the target is performed correctly. Nevertheless, even if the mean value of the RMSE shows a decrease in accuracy with respect to the case without failures, the main difference is the high dispersion of the simulations for each case. The RMSE standard deviation considerably increases, which means that, due to the stochastic nature of the failures, the convergence of one single run highly depends on when, and where, the failure occurs. Moreover, looking at Table 5.30 the number of samples that were able to converge has decreased further.

The comparison among the filters presents the same behaviour described in the simulations with no failures. However, the difference between high-order filters and the classical EKF is here more marked, especially in the ability of high order filters to be more robust and consistent showing approximately same error rate for different samples. The EKFDA1 achieves convergence, but it is more suitable to present an anomalous behaviour and, thus, it is weaker to the failure presence. All the results are reported in the appendix, while the UKF simulation results are reported here for a rapid idea on how the performance of the filters have worsen due to failure (see Table 5.25 to Table 5.28).

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	4.38e-3	1.90e-1	1.14e0	**0.05**	6.76e-6	3.46e-4	1.87e-3
0.1	1.62e-3	2.81e-2	4.36e-1	**0.1**	2.87e-6	4.25e-5	7.26e-4
0.5	6.28e-4	7.19e-4	2.77e-3	**0.5**	9.24e-7	1.09e-6	4.30e-6

Table 5.25: $_{UKF}\bar{\mu}_{3Mfail}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.73e-3	3.51e-1	6.82e-1	**0.05**	3.63e-6	7.02e-4	1.16e-3
0.1	4.59e-4	7.83e-2	6.02e-1	**0.1**	7.71e-7	1.10e-4	1.08e-3
0.5	5.88e-5	2.23e-4	9.53e-3	**0.5**	8.89e-8	2.89e-7	1.47e-5

Table 5.26: $_{UKF}\sigma_{\bar{\mu},3Mfail}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.37e-4	1.35e-3	4.21e-3	**0.05**	1.91e-5	2.96e-5	6.70e-5
0.1	4.55e-4	1.06e-3	2.47e-3	**0.1**	1.12e-5	1.87e-5	3.68e-5
0.5	9.96e-5	1.70e-4	5.30e-4	**0.5**	1.55e-6	3.91e-6	1.19e-5

Table 5.27: $_{UKF}\bar{\mu}_{3Mfail}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	2.01e-4	2.23e-4	0	**0.05**	2.97e-6	1.80e-5	0
0.1	8.16e-5	9.53e-4	1.53e-3	**0.1**	3.01e-6	1.43e-5	3.90e-5
0.5	1.97e-5	1.61e-4	7.27e-4	**0.5**	4.95e-7	4.16e-6	1.74e-5

Table 5.28: ${}_{UKF}\sigma_{\bar{\mu},3Mfail}$ sensitivity analysis for rotational performance.

EKFDA1				EKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	99	47	37	**0.05**	93	38	40
0.1	100	89	69	**0.1**	100	89	82
0.5	100	100	100	**0.5**	100	100	100
UKF				UKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	100	53	43	**0.05**	100	46	51
0.1	100	94	72	**0.1**	100	94	79
0.5	100	100	100	**0.5**	100	100	100

Table 5.29: Success rate for each combination of frequency and amplitude factor. Translational dynamics. 3 Markers limitation with failures.

EKFDA1				EKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	3	3	0	**0.05**	6	6	1
0.1	48	24	10	**0.1**	63	51	20
0.5	100	100	89	**0.5**	100	100	94
UKF				UKFDA2			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9	2	1	**0.05**	10	4	1
0.1	58	31	13	**0.1**	58	31	13
0.5	100	100	93	**0.5**	100	100	93

Table 5.30: Success rate for each combination of frequency and amplitude factor. Rotational dynamics. 3 Markers limitation with failures.

CHAPTER 6

Conclusions and Future Developments

Innovative nonlinear filters have been developed in this thesis. The filters have been proven to show better performance with respect to classical Kalman filters and to be robust to failures. The introduction of DA in the standard EKF improves the accuracy of the filter by using second expansion order terms of the Taylor series that would have been neglected. Moreover, DA improves also the numerical performance of the UKF leading to a lighter, and faster, software.

The three filters presented have been proven to achieve an accurate pose estimation of the target in the deorbit mission of ENVISAT. Moreover, the filters present higher reliability and a better accuracy level when compared to the standard EKF, especially at low acquisition measurement frequencies. The decision of using a set of markers from the image processing of the camera as the measurement model has created a realistic application. The sensitivity analysis has been performed according to the selected model and respecting the system characteristics.

Both EKFDA2 and UKFDA2 are able to capture the mean and the covariance accurately up to the second order, but EKFDA2 has a better accuracy level with respect to the UT-based filters. Anyhow, nonlinear filters are better with respect to the classic EKF. Moreover, the insertion of DA brings great benefits: EKFDA improves EKF in accuracy while UKFDA improves UKF in computational time.

This thesis opens to a variety of future developments. The marker measurement model is based on the typical measurements obtained from camera images, therefore a scenario within the filters and a real time image processing software is the next step to a more real application. Furthermore, the implemented dynamics of ENVISAT has got no couplings between the translational and rotational part, thus more complex equations that introduce flexibility can be developed and added to the system. Concerning again the dynamics, this thesis has considered the knowledge of the inertia matrix and the centre of mass position of the target as granted. A new application may consider the error in those values and aims to use the filter not only to estimate the target state, but also to identify this mismatch and try to correct it.

The UKF could be improved by trying to use higher moments knowledge, such as skewness and kurtosis, as active information to give matching weight to each sigma point [34]. In the same way, the DA-based high order filter may be improved thanks to its fast capability of calculating moments with of higher order. The higher order moments could be included in the update step of the filters to improve accuracy.

Moreover, it is worth highlighting that all the unscented transformations have been performed with the same set of parameters for the selection of the sigma points in the covariance ellipse. A statistical analysis could be done to study the UKF (and UKFDA) behaviour considering the UT parameters as variables.

Finally, this thesis has provided a comparison between two different Kalman filter types. It may be useful to extend this comparison to particle filters [10], by applying it to scenarios with multiple targets [31].

APPENDIX A

Cholesky Decomposition

In linear algebra, the Cholesky decomposition or Cholesky factorization is a decomposition of a Hermitian, positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose. Therefore, every positive definite matrix $A \in \boldsymbol{\mathcal{R}}^{n \times n}$ can be factored as $A = LL^T$ where L is lower triangular with positive diagonal terms. The complexity of computing L is $(1/3)n^3$ flops, where n is the matrix dimension. L is called the *Cholesky factor* of A and can be interpreted as the "square root" of a positive definite matrix.

The Cholesky Banachiewicz algorithm is here reported. This algorithm starts from the upper left element of matrix L and then it builds the matrix row by row.

$$l_{i,j} = \frac{1}{l_{j,j}} \left(a_{i,j} - \sum_{k=1}^{j-1} l_{i,k} l_{j,k} \right) \tag{A.1}$$

$$l_{i,j} = \sqrt{a_{i,i} - \sum_{k=1}^{i-1} l_{i,k}^2} \tag{A.2}$$

These equations applies $\forall i = 0, \ldots, n$ and $\forall j = 0, \ldots, i-1$.

APPENDIX B

Tables Results

B.1 All Markers

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	4.47e-3	1.52e-1	8.08e-1	**0.05**	6.35e-6	2.41e-4	1.39e-3
0.1	1.48e-3	1.84e-2	3.41e-1	**0.1**	2.70e-6	2.71e-5	5.53e-4
0.5	6.34e-4	6.30e-4	8.57e-4	**0.5**	9.35e-7	9.79e-7	1.54e-6
1	2.88e-4	2.81e-4	2.77e-4	**1**	5.08e-7	4.90e-7	4.68e-7
3	8.81e-5	8.95e-5	9.50e-5	**3**	1.52e-7	1.55e-7	1.65e-7

Table B.1: $_{EKFDA1}\bar{\mu}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	2.32e-3	2.82e-1	5.61e-1	**0.05**	3.61e-6	4.49e-4	1.03e-3
0.1	1.66e-4	3.30e-2	5.13e-1	**0.1**	2.52e-8	4.78e-5	9.12e-4
0.5	2.71e-6	4.84e-5	3.60e-4	**0.5**	2.79e-9	8.41e-8	7.44e-7
1	4.59e-7	7.19e-6	3.01e-5	**1**	6.62e-10	1.20e-8	4.47e-8
3	8.35e-8	1.37e-6	6.54e-6	**3**	1.35e-10	2.23e-9	1.07e-8

Table B.2: $_{EKFDA1}\sigma_{\bar{\mu}}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	8.90e-4	1.98e-3	5.13e-3	**0.05**	1.82e-5	3.48e-5	4.31e-5
0.1	4.43e-4	1.33e-3	2.64e-3	**0.1**	9.46e-6	2.51e-5	3.24e-5
0.5	9.46e-5	1.06e-4	2.27e-4	**0.5**	1.93e-6	2.31e-6	5.37e-6
1	5.16e-5	5.39e-5	6.87e-5	**1**	3.93e-7	5.30e-7	1.11e-6
3	9.44e-6	1.06e-5	1.44e-5	**3**	1.52e-7	1.95e-7	3.55e-7

Table B.3: ${}_{EKFDA1}\bar{\mu}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.49e-4	6.31e-4	3.34e-4	**0.05**	3.84e-4	1.94e-5	3.79e-5
0.1	6.96e-5	1.17e-4	2.20e-3	**0.1**	2.75e-6	2.40e-5	2.46e-5
0.5	1.81e-6	3.27e-5	2.13e-4	**0.5**	8.81e-8	1.01e-6	4.48e-6
1	1.24e-6	8.82e-6	3.27e-5	**1**	1.21e-8	2.07e-7	9.52e-7
3	1.37e-7	1.44e-6	7.30e-6	**3**	6.16e-9	6.63e-8	2.79e-7

Table B.4: ${}_{EKFDA1}\sigma_{\bar{\mu}}$ sensitivity analysis for rotational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	3.61e-3	1.53e-1	9.72e-1	**0.05**	4.68e-6	2.74e-4	1.68e-3
0.1	1.40e-3	4.60e-3	2.80e-1	**0.1**	2.60e-6	6.65e-6	4.82e-4
0.5	6.33e-4	6.29e-4	8.01e-4	**0.5**	9.34e-7	9.66e-7	1.40e-6
1	2.88e-4	2.81e-4	2.77e-4	**1**	5.08e-7	4.90e-7	4.68e-7
3	8.82e-5	8.96e-5	9.51e-5	**3**	1.52e-7	1.55e-7	1.65e-7

Table B.5: ${}_{EKFDA2}\bar{\mu}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	4.59e-5	3.62e-1	6.59e-1	**0.05**	1.15e-7	6.98e-4	1.81e-3
0.1	9.20e-6	1.18e-2	4.97e-1	**0.1**	2.43e-8	3.69e-5	9.18e-4
0.5	2.44e-6	4.59e-5	3.02e-4	**0.5**	2.61e-9	6.77e-8	6.04e-7
1	4.56e-7	7.09e-6	2.97e-5	**1**	6.55e-10	1.18e-8	4.40e-8
3	8.37e-8	1.36e-6	6.52e-6	**3**	1.35e-10	2.23e-9	1.07e-8

Table B.6: $_{EKFDA2}\sigma_{\bar{\mu}}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.36e-4	1.21e-3	2.82e-3	**0.05**	2.35e-5	2.51e-5	2.17e-5
0.1	4.49e-4	7.60e-4	1.23e-3	**0.1**	1.05e-5	1.64e-5	2.24e-5
0.5	9.46e-5	1.04e-4	2.13e-4	**0.5**	1.96e-6	2.25e-6	5.01e-6
1	5.07e-5	5.30e-5	6.79e-5	**1**	3.88e-7	5.24e-7	1.10e-6
3	9.17e-7	9.98e-6	1.41e-5	**3**	1.49e-7	1.91e-7	3.50e-7

Table B.7: $_{EKFDA2}\bar{\mu}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.54e-5	4.01e-4	2.21e-3	**0.05**	6.56e-7	6.79e-6	8.79e-6
0.1	9.47e-6	6.23e-4	1.10e-3	**0.1**	4.29e-6	1.19e-5	2.38e-5
0.5	1.75e-6	3.05e-5	1.96e-4	**0.5**	8.47e-8	9.51e-7	4.17e-6
1	1.23e-6	8.67e-6	3.24e-5	**1**	1.03e-8	2.02e-7	9.41e-7
3	1.23e-7	1.38e-6	7.25e-6	**3**	5.58e-9	6.45e-8	2.77e-7

Table B.8: $_{EKFDA2}\sigma_{\bar{\mu}}$ sensitivity analysis for rotational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	3.93e-3	1.66e-3	7.51e-1	**0.05**	5.51e-6	2.74e-4	1.33e-3
0.1	1.42e-3	1.10e-2	2.31e-1	**0.1**	2.63e-6	1.64e-5	3.83e-4
0.5	6.33e-4	6.29e-4	7.86e-4	**0.5**	9.34e-7	9.60e-7	1.34e-6
1	2.88e-4	2.81e-4	2.77e-4	**1**	5.08e-7	4.90e-7	4.68e-7
3	8.82e-5	8.96e-5	9.50e-5	**3**	1.52e-7	1.55e-7	1.65e-7

Table B.9: ${}_{UKF}\bar{\mu}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.05e-3	3.02e-1	5.37e-1	**0.05**	1.89e-6	5.35e-4	1.04e-3
0.1	6.11e-5	2.60e-2	4.17e-1	**0.1**	1.24e-7	3.66e-5	7.61e-4
0.5	2.53e-6	4.90e-5	2.64e-4	**0.5**	2.68e-9	6.39e-8	5.53e-7
1	4.53e-7	7.07e-5	2.96e-5	**1**	6.50e-10	1.18e-8	4.39e-8
3	8.37e-8	1.36e-6	6.52e-6	**3**	1.35e-10	2.25e-9	1.07e-8

Table B.10: ${}_{UKF}\sigma_{\bar{\mu}}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	8.53e-4	1.77e-3	2.75e-3	**0.05**	1.91e-5	3.06e-5	2.60e-5
0.1	4.50e-4	1.02e-3	2.12e-3	**0.1**	1.04e-5	2.00e-5	3.22e-5
0.5	9.45e-5	1.06e-4	2.23e-4	**0.5**	1.95e-6	2.32e-6	5.29e-6
1	5.07e-5	5.30e-5	6.79e-5	**1**	3.88e-7	5.24e-7	1.11e-6
3	9.17e-7	9.98e-6	1.41e-5	**3**	1.49e-7	1.91e-7	3.50e-7

Table B.11: ${}_{UKF}\bar{\mu}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	8.87e-5	1.06e-3	1.69e-3	**0.05**	3.20e-6	2.62e-5	1.80e-5
0.1	6.31e-5	7.15e-4	1.79e-3	**0.1**	2.50e-6	1.52e-5	2.45e-5
0.5	1.85e-6	3.32e-5	2.04e-4	**0.5**	8.86e-8	1.04e-6	4.42e-6
1	1.23e-6	8.70e-6	3.25e-5	**1**	1.04e-8	2.03e-7	9.45e-7
3	1.23e-7	1.38e-6	7.25e-6	**3**	5.58e-9	6.45e-8	2.77e-7

Table B.12: $_{UKF}\sigma_{\bar{\mu}}$ sensitivity analysis for rotational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	3.93e-3	1.74e-1	6.89e-1	**0.05**	5.51e-6	3.10e-4	1.24e-3
0.1	1.42e-3	1.10e-2	2.28e-1	**0.1**	2.63e-6	1.64e-5	3.63e-4
0.5	6.33e-4	6.29e-4	7.86e-4	**0.5**	9.34e-7	9.60e-7	1.34e-6
1	2.88e-4	2.81e-4	2.77e-4	**1**	5.08e-7	4.90e-7	4.68e-7
3	8.82e-5	8.96e-5	9.50e-5	**3**	1.52e-7	1.55e-7	1.65e-7

Table B.13: $_{UKFDA2}\bar{\mu}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.05e-3	3.27e-1	5.11e-1	**0.05**	1.89e-6	6.33e-4	1.02e-3
0.1	6.11e-5	2.60e-2	4.07e-1	**0.1**	1.24e-7	3.66e-5	7.15e-4
0.5	2.53e-6	4.90e-5	2.94e-4	**0.5**	2.68e-9	6.39e-8	5.53e-7
1	4.53e-7	7.07e-5	2.98e-5	**1**	6.51e-10	1.18e-8	4.39e-8
3	8.37e-8	1.36e-6	6.52e-6	**3**	1.35e-10	2.25e-9	1.07e-8

Table B.14: $_{UKFDA2}\sigma_{\bar{\mu}}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	8.54e-4	1.77e-3	2.79e-3	**0.05**	1.91e-5	3.05e-5	2.60e-5
0.1	4.50e-4	1.02e-3	2.12e-3	**0.1**	1.04e-5	2.00e-5	3.22e-5
0.5	9.45e-5	1.06e-4	2.23e-4	**0.5**	1.95e-6	2.32e-6	5.29e-6
1	5.07e-5	5.30e-5	6.79e-5	**1**	3.88e-7	5.24e-7	1.11e-6
3	9.17e-7	9.98e-6	1.41e-5	**3**	1.49e-7	1.91e-7	3.50e-7

Table B.15: ${}_{UKFDA2}\bar{\mu}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	8.86e-5	1.06e-3	1.75e-3	**0.05**	3.19e-6	2.63e-5	1.72e-5
0.1	6.31e-5	7.15e-4	1.79e-3	**0.1**	2.50e-6	1.52e-5	2.45e-5
0.5	1.85e-6	3.32e-5	2.04e-4	**0.5**	8.86e-8	1.04e-6	4.42e-6
1	1.23e-6	8.70e-6	3.25e-5	**1**	1.04e-8	2.03e-7	9.45e-7
3	1.23e-7	1.38e-6	7.25e-6	**3**	5.58e-9	6.45e-8	2.77e-7

Table B.16: ${}_{UKFDA2}\sigma_{\bar{\mu}}$ sensitivity analysis for rotational performance.

B.2 Three Markers

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	4.64e-3	1.34e-1	8.20e-1	**0.05**	6.74e-6	2.41e-4	1.28e-3
0.1	1.50e-3	2.78e-2	4.09e-1	**0.1**	2.70e-6	4.30e-5	6.66e-4
0.5	6.29e-4	6.38e-4	9.76e-4	**0.5**	9.30e-7	1.00e-6	1.74e-6
1	2.95e-4	2.89e-4	3.16e-4	**1**	5.21e-7	4.91e-7	4.83e-7
3	8.83e-5	9.22e-5	1.14e-4	**3**	1.52e-7	1.59e-7	1.92e-7

Table B.17: ${}_{EKFDA1}\bar{\mu}_{3M}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	2.89e-3	2.45e-1	6.58e-1	**0.05**	4.52e-6	5.09e-4	9.75e-4
0.1	2.24e-4	4.99e-2	5.59e-1	**0.1**	3.43e-7	7.50e-5	1.04e-3
0.5	3.50e-6	6.91e-5	4.93e-4	**0.5**	3.77e-9	1.15e-7	9.06e-7
1	9.66e-7	1.49e-5	9.38e-5	**1**	1.58e-9	2.27e-8	8.21e-8
3	2.16e-7	5.70e-6	3.87e-5	**3**	2.95e-10	7.68e-9	5.18e-8

Table B.18: ${}_{EKFDA1}\sigma_{\bar{\mu},3M}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.29e-4	2.54e-3	-	**0.05**	1.78e-5	4.15e-5	-
0.1	4.47e-4	1.05e-3	2.43e-3	**0.1**	1.17e-5	2.02e-5	2.96e-5
0.5	8.68e-5	1.08e-4	3.07e-4	**0.5**	1.30e-6	2.20e-6	7.12e-6
1	6.74e-5	7.66e-5	1.27e-4	**1**	4.36e-7	8.58e-7	2.50e-6
3	8.51e-6	1.40e-5	3.46e-5	**3**	1.30e-7	3.17e-7	9.88e-7

Table B.19: ${}_{EKFDA1}\bar{\mu}_{3M}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.43e-4	1.02e-3	-	**0.05**	5.60e-6	1.76e-5	-
0.1	8.46e-5	7.98e-4	1.65e-3	**0.1**	3.23e-6	2.11e-5	1.71e-5
0.5	3.22e-6	4.12e-5	3.42e-4	**0.5**	1.52e-7	1.36e-6	7.88e-6
1	3.03e-6	2.29e-5	9.79e-5	**1**	1.66e-8	5.68e-7	2.94e-6
3	4.48e-7	7.83e-6	3.60e-5	**3**	1.08e-8	2.63e-7	1.21e-6

Table B.20: $_{EKFDA1}\sigma_{\bar{\mu},3M}$ sensitivity analysis for rotational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.62e-2	1.89e-1	1.11e-0	**0.05**	3.23e-5	2.98e-4	1.74e-3
0.1	1.38e-3	9.62e-3	3.86e-1	**0.1**	2.65e-6	1.48e-5	6.16e-4
0.5	6.26e-4	6.34e-4	9.37e-4	**0.5**	9.27e-7	9.90e-7	1.65e-6
1	2.94e-4	2.88e-4	3.16e-4	**1**	5.22e-7	4.92e-7	4.84e-7
3	8.87e-5	9.25e-5	1.14e-5	**3**	1.53e-7	1.59e-7	1.93e-7

Table B.21: $_{EKFDA2}\bar{\mu}_{3M}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	6.83e-2	3.16e-1	6.07e-1	**0.05**	1.42e-4	5.19e-4	1.01e-3
0.1	2.23e-5	2.11e-2	5.98e-1	**0.1**	5.69e-8	3.22e-5	9.91e-4
0.5	3.33e-6	6.63e-5	4.59e-4	**0.5**	3.56e-9	1.04e-7	8.25e-7
1	9.55e-7	1.45e-5	9.24e-5	**1**	1.58e-9	2.26e-8	8.13e-8
3	2.15e-7	5.59e-6	3.83e-5	**3**	2.92e-10	7.60e-9	5.15e-8

Table B.22: $_{EKFDA2}\sigma_{\bar{\mu},3M}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.12e-4	1.76e-3	-	**0.05**	2.05e-5	2.42e-5	-
0.1	4.55e-4	8.15e-4	1.71e-3	**0.1**	1.24e-5	1.80e-5	3.14e-5
0.5	8.80e-5	1.07e-4	2.95e-4	**0.5**	1.37e-6	2.16e-6	6.83e-6
1	6.46e-5	7.34e-5	1.24e-4	**1**	4.41e-7	8.38e-7	2.45e-6
3	7.76e-6	1.31e-5	3.36e-5	**3**	1.25e-7	3.04e-7	9.68e-7

Table B.23: ${}_{EKFDA2}\bar{\mu}_{3M}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	4.42e-5	3.48e-4	-	**0.05**	2.64e-6	8.49e-6	-
0.1	1.93e-5	5.00e-4	1.58e-3	**0.1**	9.24e-7	1.27e-5	3.33e-5
0.5	3.44e-6	3.87e-5	3.23e-4	**0.5**	1.57e-7	1.29e-6	7.48e-6
1	3.07e-6	2.24e-5	9.64e-5	**1**	2.49e-8	5.34e-7	2.89e-6
3	3.50e-7	7.58e-6	3.57e-5	**3**	6.71e-9	2.54e-7	1.20e-6

Table B.24: ${}_{EKFDA2}\sigma_{\bar{\mu},3M}$ sensitivity analysis for rotational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	3.97e-3	1.50e-1	9.28e-1	**0.05**	5.80e-6	2.26e-4	1.53e-3
0.1	1.42e-3	1.66e-2	3.44e-1	**0.1**	2.61e-6	2.59e-5	5.72e-4
0.5	6.36e-4	6.34e-4	9.24e-4	**0.5**	9.27e-7	9.85e-7	1.60e-6
1	2.94e-4	2.88e-4	3.16e-4	**1**	5.22e-7	4.92e-7	4.84e-7
3	8.87e-5	9.25e-5	1.14e-4	**3**	1.53e-7	1.59e-7	1.93e-7

Table B.25: ${}_{UKF}\bar{\mu}_{3M}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.44e-3	2.97e-1	5.47e-1	**0.05**	2.79e-6	4.91e-4	1.02e-3
0.1	1.08e-4	3.25e-2	5.29e-1	**0.1**	2.01e-7	4.59e-5	9.77e-4
0.5	3.38e-6	6.85e-5	4.55e-4	**0.5**	3.69e-9	1.00e-7	7.88e-7
1	3.51e-7	1.46e-5	9.27e-5	**1**	1.57e-9	2.26e-8	8.12e-8
3	2.15e-7	5.59e-6	3.83e-5	**3**	2.92e-10	7.59e-9	5.14e-8

Table B.26: ${}_{UKF}\sigma_{\bar{\mu},3M}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.10e-4	2.21e-3	2.69e-3	**0.05**	1.89e-5	3.98e-5	4.86e-5
0.1	4.52e-4	1.06e-3	2.41e-3	**0.1**	1.20e-5	1.94e-5	3.92e-5
0.5	8.79e-5	1.08e-4	3.00e-4	**0.5**	1.37e-6	2.21e-6	6.98e-6
1	6.46e-5	7.38e-5	1.24e-4	**1**	4.41e-7	8.39e-7	2.46e-6
3	7.76e-6	1.31e-5	3.37e-5	**3**	1.24e-7	3.03e-7	9.69e-7

Table B.27: ${}_{UKF}\bar{\mu}_{3M}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.95e-5	1.40e-3	9.41e-5	**0.05**	3.06e-6	4.46e-5	2.27e-5
0.1	7.21e-5	8.90e-4	1.77e-3	**0.1**	3.17e-6	1.55e-5	3.72e-5
0.5	3.53e-6	4.04e-5	3.29e-4	**0.5**	1.60e-7	1.35e-6	7.60e-6
1	3.07e-6	2.24e-5	9.67e-5	**1**	2.48e-8	5.46e-7	2.90e-6
3	3.50e-7	7.58e-6	3.57e-5	**3**	6.72e-9	2.54e-7	1.20e-6

Table B.28: ${}_{UKF}\sigma_{\bar{\mu},3M}$ sensitivity analysis for rotational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	3.97e-3	8.93e-2	9.26e-1	**0.05**	5.80e-6	1.33e-4	1.45e-3
0.1	1.42e-3	1.66e-2	3.54e-1	**0.1**	2.61e-6	2.59e-5	5.94e-4
0.5	6.26e-4	6.34e-4	9.24e-4	**0.5**	9.27e-7	9.85e-7	1.60e-6
1	2.94e-4	2.88e-4	3.16e-4	**1**	5.22e-7	4.92e-7	4.84e-7
3	8.87e-5	9.25e-5	1.14e-4	**3**	1.53e-7	1.59e-7	1.93e-7

Table B.29: ${}_{UKFDA2}\bar{\mu}_{3M}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.44e-3	1.58e-1	5.93e-1	**0.05**	2.79e-6	2.41e-4	1.07e-3
0.1	1.08e-4	3.25e-2	5.31e-1	**0.1**	2.01e-7	4.60e-5	9.75e-4
0.5	3.38e-6	6.85e-5	4.55e-4	**0.5**	3.69e-9	1.00e-7	7.88e-7
1	9.51e-7	1.46e-5	9.27e-5	**1**	1.57e-9	2.26e-8	8.12e-8
3	2.15e-7	5.59e-6	3.83e-5	**3**	2.92e-10	7.60e-9	5.14e-8

Table B.30: ${}_{UKFDA2}\sigma_{\bar{\mu},3M}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.12e-4	2.19e-3	2.67e-3	**0.05**	1.89e-5	3.96e-5	4.82e-5
0.1	4.53e-4	1.06e-3	2.42e-3	**0.1**	1.20e-5	1.94e-5	3.91e-5
0.5	8.79e-5	1.08e-4	3.00e-4	**0.5**	1.37e-6	2.21e-6	6.98e-6
1	6.47e-5	7.38e-5	1.24e-4	**1**	4.42e-7	8.39e-7	2.46e-6
3	7.76e-6	1.31e-5	3.38e-5	**3**	1.25e-7	3.03e-7	9.69e-7

Table B.31: ${}_{UKFDA2}\bar{\mu}_{3M}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.99e-5	1.39e-3	1.32e-4	**0.05**	2.98e-6	4.46e-5	2.16e-5
0.1	7.22e-5	8.90e-4	1.77e-3	**0.1**	3.17e-6	1.55e-5	3.70e-5
0.5	3.53e-6	4.04e-5	3.29e-4	**0.5**	1.59e-7	1.35e-6	7.60e-6
1	3.07e-6	2.24e-5	9.67e-5	**1**	2.48e-8	5.46e-7	2.90e-6
3	3.50e-7	7.58e-6	3.57e-5	**3**	6.72e-9	2.54e-7	1.20e-6

Table B.32: $_{UKFDA2}\sigma_{\bar{\mu},3M}$ sensitivity analysis for rotational performance.

B.3 Measurement failures

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	7.25e-3	2.27e-1	1.05e0	**0.05**	1.08e-5	3.83e-4	1.71e-3
0.1	1.99e-3	5.33e-2	5.53e-1	**0.1**	3.46e-6	7.91e-5	9.26e-4
0.5	6.23e-4	9.20e-4	1.93e-2	**0.5**	9.29e-7	1.57e-6	2.38e-5

Table B.33: ${}_{EKFDA1}\bar{\mu}_{3Mfail}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.03e-2	3.43e-1	5.37e-1	**0.05**	1.41e-5	6.31e-4	1.04e-3
0.1	2.03e-3	1.09e-1	6.21e-1	**0.1**	3.50e-6	1.46e-4	1.11e-3
0.5	5.98e-5	1.24e-3	1.15e-1	**0.5**	8.95e-8	2.85e-6	1.48e-4

Table B.34: ${}_{EKFDA1}\sigma_{\bar{\mu},3Mfail}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	7.79e-4	2.36e-3	-	**0.05**	1.87e-5	5.24e-5	-
0.1	4.77e-4	1.41e-3	3.24e-3	**0.1**	1.16e-5	2.32e-5	4.40e-5
0.5	9.78e-5	2.18e-4	5.75e-4	**0.5**	1.46e-6	5.29e-6	1.28e-5

Table B.35: ${}_{EKFDA1}\bar{\mu}_{3Mfail}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.13e-4	1.02e-4	-	**0.05**	1.98e-6	2.61e-5	-
0.1	1.24e-4	1.16e-3	1.56e-3	**0.1**	4.10e-6	2.82e-5	4.33e-5
0.5	1.86e-5	3.12e-4	9.10e-4	**0.5**	4.77e-7	8.00e-6	2.10e-5

Table B.36: ${}_{EKFDA1}\sigma_{\bar{\mu},3Mfail}$ sensitivity analysis for rotational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.30e-2	1.95e-1	1.22e-0	**0.05**	2.43e-5	3.10e-4	1.88e-3
0.1	1.55e-3	1.61e-2	4.89e-1	**0.1**	2.77e-6	2.59e-5	8.60e-4
0.5	6.28e-4	7.18e-4	3.53e-3	**0.5**	9.24e-7	1.11e-6	4.89e-6

Table B.37: $_{EKFDA2}\bar{\mu}_{3Mfail}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	5.40e-2	3.38e-1	6.41e-1	**0.05**	1.12e-4	5.90e-4	1.06e-3
0.1	2.49e-4	3.98e-2	5.96e-1	**0.1**	4.84e-7	5.90e-5	1.10e-3
0.5	5.86e-5	2.10e-4	1.66e-2	**0.5**	8.86e-8	3.14e-7	1.94e-5

Table B.38: $_{EKFDA2}\sigma_{\bar{\mu},3Mfail}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.88e-4	1.82e-3	2.54e-3	**0.05**	2.09e-5	2.17e-5	1.69e-5
0.1	4.66e-4	1.11e-3	1.42e-3	**0.1**	1.16e-5	1.99e-5	2.22e-5
0.5	9.97e-5	1.63e-4	5.71e-4	**0.5**	1.55e-6	3.72e-6	1.19e-5

Table B.39: $_{EKFDA2}\bar{\mu}_{3Mfail}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.37e-4	7.87e-4	0	**0.05**	2.62e-6	3.85e-6	0
0.1	7.29e-5	9.21e-4	1.31e-3	**0.1**	2.76e-6	1.56e-5	1.98e-5
0.5	1.96e-5	1.46e-5	8.76e-4	**0.5**	4.93e-7	3.90e-6	1.71e-5

Table B.40: $_{EKFDA2}\sigma_{\bar{\mu},3Mfail}$ sensitivity analysis for rotational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	4.38e-3	1.90e-1	1.14e0	**0.05**	6.76e-6	3.46e-4	1.87e-3
0.1	1.62e-3	2.81e-2	4.36e-1	**0.1**	2.87e-6	4.25e-5	7.26e-4
0.5	6.28e-4	7.19e-4	2.77e-3	**0.5**	9.24e-7	1.09e-6	4.30e-6

Table B.41: ${}_{UKF}\bar{\mu}_{3Mfail}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.73e-3	3.51e-1	6.82e-1	**0.05**	3.63e-6	7.02e-4	1.16e-3
0.1	4.59e-4	7.83e-2	6.02e-1	**0.1**	7.71e-7	1.10e-4	1.08e-3
0.5	5.88e-5	2.23e-4	9.53e-3	**0.5**	8.89e-8	2.89e-7	1.47e-5

Table B.42: ${}_{UKF}\sigma_{\bar{\mu},3Mfail}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.37e-4	1.35e-3	4.21e-3	**0.05**	1.91e-5	2.96e-5	6.70e-5
0.1	4.55e-4	1.06e-3	2.47e-3	**0.1**	1.12e-5	1.87e-5	3.68e-5
0.5	9.96e-5	1.70e-4	5.30e-4	**0.5**	1.55e-6	3.91e-6	1.19e-5

Table B.43: ${}_{UKF}\bar{\mu}_{3Mfail}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	2.01e-4	2.23e-4	0	**0.05**	2.97e-6	1.80e-5	0
0.1	8.16e-5	9.53e-4	1.53e-3	**0.1**	3.01e-6	1.43e-5	3.90e-5
0.5	1.97e-5	1.61e-4	7.27e-4	**0.5**	4.95e-7	4.16e-6	1.74e-5

Table B.44: ${}_{UKF}\sigma_{\bar{\mu},3Mfail}$ sensitivity analysis for rotational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	4.39e-3	1.69e-1	1.15e0	**0.05**	6.76e-6	2.73e-4	1.86e-3
0.1	1.62e-3	2.78e-2	3.92e-1	**0.1**	2.87e-6	4.22e-5	6.44e-4
0.5	6.28e-4	7.19e-4	2.77e-3	**0.5**	9.24e-7	1.09e-6	4.30e-6

Table B.45: $_{UKFDA2}\bar{\mu}_{3Mfail}$ sensitivity analysis for translational performance.

Relative Position (m)				Relative Velocity (m/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.73e-3	3.31e-1	6.32e-1	**0.05**	3.66e-6	5.58e-4	1.07e-3
0.1	4.60e-4	7.62e-2	5.06e-1	**0.1**	7.75e-7	1.08e-4	9.15e-4
0.5	5.88e-5	2.23e-4	9.53e-3	**0.5**	8.89e-8	2.89e-7	1.47e-5

Table B.46: $_{UKFDA2}\sigma_{\bar{\mu},3Mfail}$ sensitivity analysis for translational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	9.54e-4	1.14e-3	3.26e-3	**0.05**	1.88e-5	2.21e-5	3.55e-5
0.1	4.55e-4	1.06e-3	2.46e-3	**0.1**	1.12e-5	1.87e-5	3.75e-5
0.5	9.96e-5	1.70e-4	5.30e-4	**0.5**	1.55e-6	3.91e-6	1.19e-5

Table B.47: $_{UKFDA2}\bar{\mu}_{3Mfail}$ sensitivity analysis for rotational performance.

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
0.05	1.97e-4	2.98e-4	0	**0.05**	3.03e-6	1.03e-5	0
0.1	8.15e-5	9.50e-4	1.45e-3	**0.1**	3.01e-6	1.43e-5	2.87e-5
0.5	1.97e-5	1.61e-4	7.27e-4	**0.5**	4.95e-7	4.16e-6	1.74e-5

Table B.48: $_{UKFDA2}\sigma_{\bar{\mu},3Mfail}$ sensitivity analysis for rotational performance.

Bibliography

[1] European Space Agency. Annex c: Envisat specification. pages 22–23, 17/05/2016.

[2] R. Armellin et al. Alezander Witting, P. Di Lizia. An introduction to differential algebra. Dinamica Innovating Technology, European Space Agency.

[3] Roberto Armellin, Pierluigi Di Lizia, Franco Bernelli-Zazzera, and M Berz. Asteroid close encounters characterization using differential algebra: the case of apophis. *Celestial Mechanics and Dynamical Astronomy*, 107(4):451–470, 2010.

[4] M Berz. High-order computation and normal form analysis of repetitive systems. In *AIP Conference Proceedings*, volume 249, pages 456–489. AIP, 1992.

[5] M Berz. Differential algebraic techniques, entry in handbook of accelerator physics and engineering. M. Tinger and A. Chao (Eds.), World Scientific, 1999.

[6] Martin Berz. The new method of tpsa algebra for the description of beam dynamics to high orders. *TechnicalReport AT-6: ATN-86-16, Los Alamos National Laboratory*, 1986.

[7] Martin Berz. The method of power series tracking for the mathematical description of beam dynamics. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 258(3):431–436, 1987.

[8] Martin Berz. *Advances in Imaging and Electron Physics.* Academic Press, 1999.

[9] Robin Biesbroek, Tiago Soares, Jakob Husing, and Luisa Innocenti. The e. deorbit cdf study: a design study for the safe removal of a large space debris. In *6th European Conference on Space Debris*, volume 723, 2013.

[10] Eleni N Chatzi and Andrew W Smyth. The unscented kalman filter and particle filter methods for nonlinear structural system identification with non-collocated heterogeneous sensing. *Structural control and health monitoring*, 16(1):99–123, 2009.

[11] John L Crassidis and John L Junkins. *Optimal estimation of dynamic systems*. CRC press, 2011.

[12] John L Crassidis and F Landis Markley. Attitude estimation using modified rodrigues parameters. 1996.

[13] John L Crassidis and F Landis Markley. Sliding mode control using modified rodrigues parameters. *Journal of Guidance, Control and Dynamics*, 19(6):1, 1996.

[14] John L Crassidis and F Landis Markley. Unscented filtering for spacecraft attitude estimation. *Journal of Guidance Control and Dynamics*, 26(4):536–542, 2003.

[15] JFG De Freitas, Mahesan Niranjan, Andrew Gee, and Arnaud Doucet. Sequential monte carlo methods for optimisation of neural network models. *Cambridge University Engineering Department, Cambridge, England, Technical Report TR-328*, 1998.

[16] Pierluigi Di Lizia, Mauro Massari, and Francesco Cavenago. Assessment of onboard da state estimation for spacecraft relative navigation. 2017.

[17] Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. cambridge University press, 2009.

[18] Leon Isserlis. On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika*, 12(1/2):134–139, 1918.

[19] Simon Julier, Jeffrey Uhlmann, and Hugh F Durrant-Whyte. Technical notes and correspondence. *IEEE Transactions On Automatic Control*, 45(3):477, 2000.

[20] Simon J Julier. The scaled unscented transformation. In *American Control Conference, 2002. Proceedings of the 2002*, volume 6, pages 4555–4559. IEEE, 2002.

[21] Simon J Julier and Jeffrey K Uhlmann. A general method for approximating nonlinear transformations of probability distributions. Technical report, Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, 1996.

[22] Simon J Julier and Jeffrey K Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls*, volume 3, pages 182–193. Orlando, FL, 1997.

[23] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[24] Jed M Kelsey, Jeffrey Byrne, Martin Cosgrove, Sanjeev Seereeram, and Raman K Mehra. Vision-based relative pose estimation for autonomous rendezvous and docking. In *Aerospace Conference, 2006 IEEE*, pages 20–pp. IEEE, 2006.

[25] Son-Goo Kim, John L Crassidis, Yang Cheng, Adam M Fosbury, and John L Junkins. Kalman filtering for relative spacecraft attitude and position estimation. *Journal of Guidance Control and Dynamics*, 30(1):133–143, 2007.

[26] Steven J Leon. *Linear algebra with applications.* Macmillan New York, 1980.

[27] Pierluigi Di Lizia. *Robust Space Trajectory and Space System Design using Differential Algebra.* Phd thesis in aerospace engineering, XX Ciclo, March 2008.

[28] Berz M. *Modern Map Methods in Particle Beam Physics.* Academic Press, 1999.

[29] Manoranjan Majji, James D Turner, and John L Junkins. A high order method for estimation of dynamic systems part i: Theory1. *differential equations*, 1(1):1–1, 2008.

[30] Kyoko Makino and Martin Berz. Cosy infinity version 9. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 558(1):346–350, 2006.

[31] M Marron, JC Garcia, MA Sotelo, M Cabello, D Pizarro, F Huerta, and J Cerro. Comparing a kalman filter and a particle filter in a multiple objects tracking application. In *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, pages 1–6. IEEE, 2007.

[32] Ryan S Park and Daniel J Scheeres. Nonlinear mapping of gaussian statistics: theory and applications to spacecraft trajectory design. *Journal of Guidance Control and Dynamics*, 29(6):1367–1375, 2006.

[33] Ryan S Park and Daniel J Scheeres. Nonlinear semi-analytic methods for trajectory estimation. *Journal of Guidance Control and Dynamics*, 30(6):1668, 2007.

[34] Ksenia Ponomareva, Paresh Date, and Zidong Wang. A new unscented kalman filter with higher order moment-matching. *Proceedings of Mathematical Theory of Networks and Systems (MTNS 2010), Budapest*, 2010.

[35] Mirco Rasotto, Alessandro Morselli, Alexander Wittig, Mauro Massari, Pierluigi Di Lizia, Roberto Armellin, Celia Valles, and Guillermo Ortega. Differential algebra space toolbox for nonlinear uncertainty propagation in space dynamics. 2016.

[36] Bob Schutz, Byron Tapley, and George H Born. *Statistical orbit determination.* Academic Press, 2004.

[37] MD Shuster, EJ Lefferts, and FL Markley. Kalman filtering for spacecraft attitude estimation. In *AIAA 20th Aerospace Sciences Meeting, Orlando, Florida*, volume 232, 1982.

[38] Tsu T Soong. *Fundamentals of probability and statistics for engineers.* John Wiley & Sons, 2004.

[39] Cyrill Stachniss. Robot mapping - unscented kalman filter. Uni Freiburg - ASI: Autonomous Intelligence Systems.

[40] Hisashi Tanizaki. *Nonlinear filters: estimation and applications.* Springer Science & Business Media, 2013.

[41] JK UhlmannS, J Julier, and HF Durrant-Whyte. A new approach for the nonlinear transformation of means and covariances in linear filters. *IEEE Trans. Automatic Control*, 1996.

[42] M Valli, R Armellin, P Di Lizia, and MR Lavagna. Nonlinear mapping of uncertainties in celestial mechanics. *Journal of Guidance, Control, and Dynamics*, 2013.

[43] Monica Valli, Roberto Armellin, Pierluigi Di Lizia, and Michèle R Lavagna. Nonlinear filtering methods for spacecraft navigation based on differential algebra. *Acta Astronautica*, 94(1):363–374, 2014.

[44] Rudolph Van Der Merwe. Sigma-point kalman filters for probabilistic inference in dynamic state-space models. 2004.

[45] James R Van Zandt. A more robust unscented transform. In *International symposium on optical science and technology*, pages 371–380, 2001.

[46] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. Ieee, 2000.

[47] Han Wang and Michael Brady. Real-time corner detection algorithm for motion estimation. *Image and vision computing*, 13(9):695–703, 1995.

[48] Eric W Weisstein. Rotation matrix. 2003.

[49] J Wertz and Robert Bell. Autonomous rendezvous and docking technologiesstatus and prospects. In *SPIEs 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls,(Orlando, USA)*, pages 21–25, 2003.

[50] Alexander Wittg. Differential algebra lab. In *Astronet II - Fourth Training School*, February 2015.

[51] Kjetil Wormnes, Ronan Le Letty, Leopold Summerer, Rogier Schonenborg, Olivier Dubois-Matra, Eleonora Luraschi, Alexander Cropp, Holger Krag, and Jessica Delaval. Esa technologies for space debris remediation. In *Proceedings of the 6th IAASS Conference: Safety is Not an Option*, pages 3–4, 2013.

[52] Wan-jin Zhao, Sheng-rong Gong, Chun-ping Liu, and Xiang-jun SHEN. Adaptive harris corner detection algorithm. *Computer Engineering*, 10(5):212–215, 2008.