S. Servadio, F. Cavenago, P. Di Lizia, M. Massari
Nonlinear Prediction in Marker-Based Spacecraft Pose Estimation with Polynomial Transition Maps
Journal of Spacecraft and Rockets, Vol. 59, N. 2, 2022, p. 511-523
doi:10.2514/1.A35068

Nonlinear Prediction in Marker-Based Spacecraft Pose Estimation with Polynomial Transition Maps

Simone Servadio*
Massachusetts Institute of Technology, Cambridge, 02139, MA, USA

Francesco Cavenago† and Pierluigi Di Lizia‡ and Mauro Massari§
Politecnico Di Milano, Milano 20156, Italy

Spacecraft relative state estimation is of paramount importance in the problem of rendezvous with an uncooperative target; indeed, an accurate prediction of its relative position and attitude are crucial for safe proximity operations, especially considering autonomous guidance, navigation and control. Therefore, a key point for the success of these missions is the development of efficient algorithms capable of limiting the computational burden without any reduction in performance. This paper addresses the issue proposing and analyzing nonlinear filters based on differential algebra (DA). High-order numerical extended Kalman filter and unscented Kalman filter are developed in the DA framework and their performance is assessed and compared in terms of accuracy, robustness and computational time, highlighting advantages and drawbacks. The European Space Agency e.deorbit mission, involving Envisat, is considered as reference case and the analysis is carried out through numerous numerical simulations, taking into account different measurement acquisition frequency and level of uncertainties.

Nomenclature

$\mathbf{\Gamma}$	=	Rotation Matrix
μ	=	Gravitational Parameter
$\boldsymbol{\zeta}$	=	Modified Rodriguez Parameters
$\boldsymbol{\rho}$	=	Marker Position w.r.t. the center of mass
σ_r	=	Position Standard Deviation
σ_v	=	Velocity Standard Deviation
$\boldsymbol{\omega_c}$	=	Chaser angular velocity

An initial version of this paper was presented at the 2018 Space Flight Mechanics Meeting, 8-12 January 2018, Kissimmee, Florida. Paper number AIAA 2018-1964

*Postdoctoral Associate, Dept of of Aeronautics and Astronautics, simoserv@mit.edu
†Postdoctoral Researcher, Dept of Aerospace Science and Technologies, francesco.cavenago@polimi.it
‡Assistant Professor, Dept of Aerospace Science and Technologies, pierluigi.dilizia@polimi.it
§Associate Professor, Dept of Aerospace Science and Technologies, mauro.massari@polimi.it

$\boldsymbol{\omega_r}$ = Relative angular velocity

$\boldsymbol{\omega_t}$ = Target angular velocity

$\boldsymbol{J_t}$ = Inertia Matrix of the Target

$\boldsymbol{K}$ = Kalman gain

$\boldsymbol{m^-}$ = Predicted State Mean

$\boldsymbol{m^+}$ = Updated State Mean

$\boldsymbol{M_app}$ = Apparent Torques

$\boldsymbol{M_ci}$ = Chaser-Inertial Torques

$\boldsymbol{M_g}$ = Gyroscopic Torques

$\boldsymbol{n^-}$ = Predicted Measurement Mean

$\boldsymbol{P^-}$ = Predicted State Covariance

$\boldsymbol{P^+}$ = Updated State Covariance

$\boldsymbol{P^{xz}}$ = State and Measurement Predicted Cross-Covariance

$\boldsymbol{P^{zz}}$ = Measurement Predicted Covariance

$\boldsymbol{q}$ = Quaternions

$\boldsymbol{Q}$ = Process Noise Covariance

$\boldsymbol{r}$ = Spacecraft relative position

$\bar{\boldsymbol{r}}$ = Earth-chaser distance

$\boldsymbol{R}$ = Measurement Noise Covariance

$\boldsymbol{v}$ = Spacecraft relative velocity

$\boldsymbol{x}$ = State of the system

$\mathcal{X}$ = State Sigma Points Matrix

$\boldsymbol{w}$ = Sigma Points weights

$\boldsymbol{z}$ = Measurement Outcome

$\mathcal{Z}$ = Measurement Sigma Points Matrix

I. Introduction

Missions like active debris removal, on-orbit servicing, satellite inspection have gained increasing importance in the space community due to the necessity of reducing costs and guaranteeing a safer environment. In this context, autonomous guidance, navigation and control (GNC) plays a fundamental role in the problem of rendezvous with an uncooperative target. Especially, the estimation of the relative position and the prediction of the target attitude are crucial for safe proximity operations [1]. This requires complex computations to be executed on-board with a frequency

that guarantees the fulfillment of the accuracy requirements. Unfortunately, current space processors have limited computational power which restricts the implementable estimation processes. Therefore, a key point for the success of these missions is the development of efficient algorithms capable of limiting the computational burden without losing out the necessary performance. To this aim, this study analyzes the exploitation of nonlinear filters based on differential algebra (DA). Especially, high-order numerical extended Kalman filter (HNEKF) and unscented Kalman filter (UKF) are implemented in the DA framework.

Many filtering techniques have been developed to deal with estimation problems. At present time, one of the most exploited estimation algorithm is the extended Kalman filter (EKF) [2]. The EKF is based on the main idea of linearizing the equations of motion and the measurement equations via first-order Taylor expansions around the current mean and covariance (the uncertainties are assumed to be Gaussian distributed). In some cases, however, the linear assumption may fail due to the nature of the dynamics or the number of available measurements, leading to inaccurate realization of the local motion. Therefore, alternative methods capable of accounting for system nonlinearity must be used. A different approach is the UKF [3, 4]. This technique is based on the unscented transformation, which does not contain any linearization. Carefully-chosen sample points are propagated through the true nonlinear system in order to capture the posterior mean and covariance. Thanks to its nonlinear nature, UKF provides superior performance with respect to the EKF in highly nonlinear situations. In 2007, Park and Scheeres [5, 6] developed the HNEKF by implementing a semi-analytic orbit uncertainty propagation technique, that is by solving for the higher-order Taylor series terms, that describe the localized nonlinear motion, and by analytically mapping the initial uncertainties. These higher-order filters are more accurate than the EKF, as the prediction step relies on a fully nonlinear mapping of the means and covariances. However, the HNEKF needs to derive the so-called higher-order tensors, which makes it in many cases difficult to use due to computational complexity. Majji, Turner, and Junkins [7], on the other hand, expand the work by Park and Scheeres to include measurement updates in all the higher order central moments. They develop a new tensorial mechanization to exploit Object Oriented Coordinate Embedding method (OCEA) to evaluate desired order state transition tensors (STT) automatically. However, their technique still relies on the evaluation of STTs, and the relative storage.

The complexity of integrating multiple points, for UKF, and of deriving the high-order tensor, for the HNEKF, to map the mean and covariance can be easily solved using DA techniques. By substituting the classical implementation of real algebra with the implementation of a new algebra of Taylor polynomials, any function f of n variables can be easily expanded into its Taylor polynomial up to an arbitrarily order m in the DA framework [8, 9]. 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. Consequently, by propagating the mean trajectory and evaluating the measurement function in the DA framework, not only their pointwise values are obtained, but also the higher-order partials. This eliminates the need to calculate the higher-order tensors at each time step by solving a complex system of augmented ODE, for the HNEKF [10–12]. Moreover, it reduces the multiple integration of the UKF to an easier evaluation of the Taylor expansion of the flow in different sample points.

The European Space Agency (ESA) e.deorbit mission [13], involving the Envisat satellite, is used as reference test case for the developed filters. For the filtering, the measurements are given by a camera that acquires the position of markers on the target and, through image processing, provides information about the relative position and attitude. Both HNEKF and UKF are applied to this target application and compared in terms of accuracy and computational burden. The performance are assessed through a sensitivity analysis to the measurement acquisition frequency and level of uncertainties in the system. Moreover, occurrence of failures in the markers recognition is taken into account.

The main contributions of this work are: 1) the development of the UKF in the DA framework for the first time, which results in a new filter as accurate as the standard UKF, but computationally lighter; 2) a more comprehensive study with respect to [12] on the application of DA-based filters to the relative pose estimation problem, which includes the performance assessment of the new DA-based UKF, the introduction of nonlinear coupled measurement equations and a robustness analysis considering a variable number of available measurements and even the lack of them; 3) a performance comparison among the DA-based filters, highlighting their strengths and weaknesses for the application case.

The paper is organized as follows. First, an introduction to DA is given and the derivation of the DA-based HNEKF and DA-based UKF are explained. Afterwards, the considered relative pose estimation problem is introduced and the dynamics and measurement models are developed. Finally, the performance of the filters are assessed through numerical simulations.

II. Differential algebra

Differential Algebra techniques allow solving analytical problems through an algebraic approach [9]. Similar to the computer representation of real numbers as Floating Point (FP) numbers, DA allows the representation and manipulation of functions on a computer. Each sufficiently often differentiable function f is represented by its Taylor expansion around an expansion point truncated at an arbitrary finite order. Without loss of generality, 0 is chosen as the expansion point. Algebraic operations on the space of truncated Taylor polynomials are defined such that they approximate the operations on the function space $C^r(0)$ of r times differentiable functions at 0. More specifically, each operation is defined to result in the truncated Taylor expansion of the correct result computed on the function space $C^r(0)$. This yields the so-called Truncated Power Series Algebra (TPSA) [8].

In a computer environment, each DA variable can be represented as a table of coefficients and exponents, while

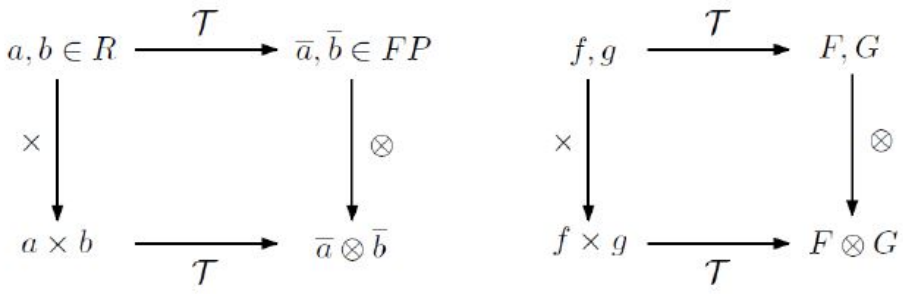

Fig. 1 Analogy between the floating point representation of real numbers in a computer environment (left figure) and the introduction of the algebra of Taylor polynomials in the differential algebraic framework (right figure)

each floating-point number is just, indeed, a number. Consider Fig. 1, which depicts the analogy between the floating point (FP) representation of real number in a computer and the algebra of Taylor polynomials in the DA framework. Take two real numbers a and b: in order to operate in a computer environment, they are represented by their FP representation, $\bar{\mathbf{a}}$ and $\bar{\mathbf{b}}$ respectively. Then, given any operation "$\times$" in the set of the real numbers, an adjoint operation "$\otimes$" is defined in the set of the FP numbers such that the diagram in Fig. 1 commutes. Consequently, the same result is obtained either if the real numbers a and b are represented in their FP representation prior to operating on them, or if the operation is carried out between the real numbers and subsequently the result expressed in its FP representation (refer to the left portion of Fig. 1). Analogously, starting from two different, sufficiently regular, functions f and g, using the DA framework, the computer environment operates on them by representing them with their Taylor series expansions, referred respectively as F and G. Similarly to the representation of real numbers with their FP expression, the functions f and g are now represented with their truncated Taylor series expansion up to a user-selected order c. For each operation in the space of function, an adjoint operation in the space of Taylor polynomials is defined such that the diagram in Fig. 1 commutes. As for real numbers, the same end result is obtain either by extracting the Taylor expansions of f and g and operating on them in the set of Taylor polynomial, or by operating on f and g in the original space and then extracting the Taylor expansion of the resulting function. To illustrate the process more clearly, consider Fig. 2. The expression $1/(x+1)$ is evaluated once in $C^r(0)$ (top) and then in DA with truncation order 3. Starting with the identity function x, one is summed to arrive at the function $x+1$, the representation of which is fully accurate in DA as it is a polynomial of order 1. Continuing the evaluation, the multiplicative inversion is performed, resulting in the function $1/(1+x)$ in $C^r(0)$. As this function is not a polynomial any more, it is automatically approximated in DA arithmetic by its truncated Taylor expansion around 0, given by $1 - x + x^2 - x^3$. As already explained, by definition of the DA operations, the diagram for each single operation commutes. That is to say the same result is reached by first Taylor expanding a $C^r(0)$ function (moving from the top to the bottom of the diagram) and then performing the DA operation (moving from left to right), or by first performing the $C^r(0)$ operation and then Taylor expanding the result.

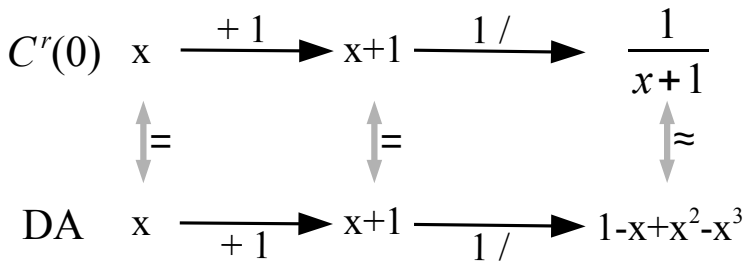

Fig. 2 Evaluation of the expression $1/(1+x)$ in $C^r(0)$ and DA arithmetic.

In addition to algebraic operations, the DA framework can be endowed with natural differentiation and integration operators, completing the structure of a differential algebra. Intrinsic functions, such as trigonometric and exponential functions, are built from elementary algebraic operations [9]. In this way, Taylor expansions of arbitrary sufficiently smooth functions given by some closed-form expression can be computed fully algebraically in a computer environment. An implementation of such DA computer routines is available in the software DACE 2.0 *, which is used to implement the algorithms presented in this paper.

An important application of DA in engineering applications is the expansion of the flow $\varphi(t; x_0)$ of an Ordinary Differential Equation (ODE) to arbitrary order with respect to initial conditions, integration times and system parameters [14]. The following is a short summary of the underlying concept. For a more complete introduction to DA, as well as a fully worked out illustrative example of a DA based ODE integrator using a simple Euler step, see [10].

Consider the initial value problem

$$\begin{cases} \dot{x} = f(x,t) \\ x(t_0) = x_0, \end{cases} \tag{1}$$

and its associated flow $\varphi(t; x_0)$. By means of classical numerical integration schemes, such as Runge-Kutta or multi-step methods, it is possible to compute the orbit of a single initial condition x_0 using floating point arithmetic on a computer. Starting instead from the DA representation of an initial condition x_0, and performing all operations in the numerical integration scheme in DA arithmetic, DA allows propagating the Taylor expansion of the flow around x_0 forward in time, up to the desired final time t_f, yielding a polynomial expansion of $\varphi(t_f; x_0 + \delta x_0)$ up to arbitrary order. Indeed, at each time step t_i, DA creates a polynomial transition map that approximate the flow of a dynamical system up to an arbitrary order. In the case of selecting first order, the coefficients of the transition map are the entries of the State Transition Matrix (STM).

The conversion of standard explicit integration schemes to their DA counterparts is rather straightforward. One simply replaces all operations performed during the execution of the scheme by the corresponding DA operations. Step size control and error estimates are performed only on the constant part of the polynomial, i.e. the reference trajectory

*`https://github.com/dacelib/dace`, accessed on 22 May 2019

of the expansion point. The result is an automatic Taylor expansion of the result of the numerical method (i.e. the numerical approximation to the flow) with respect to any quantity that was initially set to a DA value.

The main advantage of the DA-based approach is that there is no need to derive, implement and integrate variational equations in order to obtain high-order expansions of the flow. As this is achieved by merely replacing algebraic operations on floating-point numbers by DA operations, the method is inherently ODE independent. Furthermore, an efficient implementation of DA such as the DACE 2.0 package, allows to obtain high-order expansions with limited computational time.

III. High-order extended Kalman filter

This section is devoted to introducing the algorithm of the DA-based HNEKF and to provide a first assessment of its performance. The equations of motion and measurement equations describing a generic dynamic system are as follows:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{\Phi}(t_{k+1}; \boldsymbol{x}_k, t_k) + \boldsymbol{w}_k, \\ \boldsymbol{z}_{k+1} &= \boldsymbol{h}(\boldsymbol{x}_{k+1}, t_{k+1}) + \boldsymbol{v}_{k+1}, \end{aligned} \tag{2}$$

where $\boldsymbol{x}_k$ is the m-dimensional vector of state, $\boldsymbol{w}_k$ is the process noise perturbing the state, z_k is the n-dimensional vector of actual measurements, $\boldsymbol{h}$ is the measurement function, and $\boldsymbol{v}_{k+1}$ is the measurement noise characterizing the observation error. The process noise and the measurement noise are assumed to be uncorrelated, that is, $E\{\boldsymbol{v}_i\ \boldsymbol{w}_j^T\} = 0$, with the autocorrelations $E\{\boldsymbol{w}_i\ \boldsymbol{w}_j^T\} = \boldsymbol{Q}_i\delta_{ij}$ and $E\{\boldsymbol{v}_i\ \boldsymbol{v}_j^T\} = \boldsymbol{R}_i\delta_{ij}$ for all discrete time indexes i and j. $E\{\}$ denotes the expectation operator.

Starting from the general theory of state estimation, HNEKF sequentially estimate the spacecraft state and the associated uncertainty by incorporating system nonlinearity in terms of higher-order Taylor expansions and relying on the assumption that uncertainties can be described using Gaussian statistics.

Consider the system model equations (2). The filtering process can be summarized as follows:

1) *Prediction step*: at time t_{k+1}, the mean and covariance of the state vector, $\boldsymbol{m}_{k+1}^-$ and $\boldsymbol{P}_{k+1}^-$, and the mean of the measurements, $\boldsymbol{n}_{k+1}^-$, are estimated as:

$$\begin{aligned} m_{k+1,i}^- &= E\{\Phi_i(t_{k+1}; \boldsymbol{x}_k, t_k) + w_{k,i}\}, \\ P_{k+1,ij}^- &= E\{[\Phi_i(t_{k+1}; \boldsymbol{x}_k, t_k) - m_{k+1,i}^- + w_{k,i}][\Phi_j(t_{k+1}; \boldsymbol{x}_k, t_k) - m_{k+1,j}^- + w_{k,j}]\}, \\ n_{k+1,p}^- &= E\{h_p(\boldsymbol{x}_{k+1}, t_{k+1}) + v_{k+1,p}\}, \end{aligned} \tag{3}$$

where $i, j = 1, ..., m$, $p = 1, ..., n$, and $m_{k+1,i}^-$, $P_{k+1,ij}^-$ and $n_{k+1,l}^-$ are the components of $\boldsymbol{m}_{k+1}^-$, $\boldsymbol{P}_{k+1}^-$, and $\boldsymbol{n}_{k+1}^-$ respectively;

2) *Update step*: the new measurements acquired at time t_{k+1}, z_{k+1}, are incorporated into the updated estimate of the state vector and covariance matrix as follows:

$$\begin{aligned}
P^{zz}_{k+1,pq} &= E\{[h_p(\boldsymbol{x}_{k+1}, t_{k+1}) - n^-_{k+1,p} + v_{k+1,p}][h_p(\boldsymbol{x}_{k+1}, t_{k+1}) - n^-_{k+1,q} + v_{k+1,p}]\},\\
P^{xz}_{k+1,ip} &= E\{[\Phi_i(t_{k+1}; \boldsymbol{x}_k, t_k) - m^-_{k+1,i} + w_{k,i}][h_p(\boldsymbol{x}_{k+1}, t_{k+1}) - n^-_{k+1,p} + v_{k+1,p}]\},\\
\boldsymbol{K}_{k+1} &= \boldsymbol{P}^{xz}_{k+1}(\boldsymbol{P}^{zz}_{k+1})^{-1},\\
\boldsymbol{m}^+_{k+1} &= \boldsymbol{m}^-_{k+1} + \boldsymbol{K}_{k+1}(z_{k+1} - \boldsymbol{n}^-_{k+1}),\\
\boldsymbol{P}^+_{k+1} &= \boldsymbol{P}^-_{k+1} - \boldsymbol{K}_{k+1}\boldsymbol{P}^{zz}_{k+1}\boldsymbol{K}^T_{k+1},
\end{aligned} \tag{4}$$

where $q = 1, ..., n$, K_{k+1} is the Kalman gain matrix, P^{xz}_{k+1} is the cross-covariance matrix of the state and the measurement, and P^{zz}_{k+1} is the covariance matrix of the measurements.

A. The DA-based HNEKF

The DA implementation of the HNEKF relies on the fact that DA can easily provide the arbitrary order Taylor expansion of both $\boldsymbol{\Phi}$ and $\boldsymbol{h}$ in Eq. (2). Thus, the arbitrary order expansion of the equations of motion and measurement equations can be easily written, and component-wise reads:

$$\begin{aligned}
x_{k+1,i} &= \Phi_i(t_{k+1}; \boldsymbol{m}^+_k, t_k) + \textstyle\sum_{r=1}^{v} \frac{1}{r!}\Phi^{i,\gamma_1\dots\gamma_r}_{(t_{k+1},t_k)}\delta x^{\gamma_1}_{k,1}\dots\delta x^{\gamma_r}_{k,m} + w_{k,i},\\
z_{k+1,p} &= h_p(\boldsymbol{\Phi}(t_{k+1}; \boldsymbol{m}^+_k, t_k), t_{k+1}) + \textstyle\sum_{r=1}^{v} \frac{1}{r!}h^{p,\gamma_1\dots\gamma_r}_{(t_{k+1},t_k)}\delta x^{\gamma_1}_{k,1}\dots\delta x^{\gamma_r}_{k,m} + v_{k+1,p},
\end{aligned} \tag{5}$$

where v is the order of the expansion, $\gamma_i \in \{1, ..., m\}$, $\Phi^{i,\gamma_1\dots\gamma_r}_{(t_{k+1},t_k)}$ includes the higher-order partials of the solution flow, which map the deviations at time k to time $k+1$, and $h^{p,\gamma_1\dots\gamma_r}_{(t_{k+1},t_k)}$ includes the higher-order partials of the measurement function. Both $\Phi^{i,\gamma_1\dots\gamma_r}_{(t_{k+1},t_k)}$ and $h^{p,\gamma_1\dots\gamma_r}_{(t_{k+1},t_k)}$ are obtained directly by integrating the equations of motion and evaluating the measurement equations in the DA framework.

The Taylor polynomials of Eq. (5) can be inserted into Eqs. (3) and (4) to obtain the steps of the high-order extended Kalman filter:

1) *Prediction step*: at time t_{k+1}, the mean and covariance of the state vector, $\boldsymbol{m}^-_{k+1}$ and $\boldsymbol{P}^-_{k+1}$, and the mean of the

measurements, $\boldsymbol{n}_{k+1}^{-}$, are estimated as:

$$\begin{aligned}
m_{k+1,i}^{-} &= \Phi_i(t_{k+1}; \boldsymbol{m}_k^{+}, t_k) + \sum_{r=1}^{v} \frac{1}{r!} \Phi_{(t_{k+1},t_k)}^{i,\gamma_1 \ldots \gamma_r} E\{\delta x_{k,1}^{\gamma_1} \ldots \delta x_{k,m}^{\gamma_r}\}, \\
P_{k+1,ij}^{-} &= \sum_{r=1}^{v} \sum_{s=1}^{v} \frac{1}{r!s!} \Phi_{(t_{k+1},t_k)}^{i,\gamma_1 \ldots \gamma_r} \Phi_{(t_{k+1},t_k)}^{j,\xi_1 \ldots \xi_s} E\{\delta x_{k,1}^{\gamma_1} \ldots \delta x_{k,m}^{\gamma_r} \delta x_{k,1}^{\xi_1} \ldots \delta x_{k,m}^{\xi_s}\} + \\
&\quad -\delta m_{k+1}^{i} \delta m_{k+1}^{j} + Q_k^{ij}, \\
n_{k+1,p}^{-} &= h_p(\boldsymbol{\Phi}(t_{k+1}; \boldsymbol{m}_k^{+}, t_k), t_{k+1}) + \sum_{r=1}^{v} \frac{1}{r!} h_{(t_{k+1},t_k)}^{p,\gamma_1 \ldots \gamma_r} E\{\delta x_{k,1}^{\gamma_1} \ldots \delta x_{k,m}^{\gamma_r}\},
\end{aligned} \tag{6}$$

where $\xi_i \in \{1, ..., m\}$ and $\delta m_{k+1}^{i} = \Phi_i(t_{k+1}; \boldsymbol{m}_k^{+}, t_k) - m_{k+1,i}^{-}$;

2) *Update step*: the new measurements acquired at time t_{k+1}, z_{k+1}, are incorporated into the updated estimate of the state vector and covariance matrix as follows:

$$\begin{aligned}
P_{k+1,pq}^{zz} &= \sum_{r=1}^{v} \sum_{s=1}^{v} \frac{1}{r!s!} h_{(t_{k+1},t_k)}^{p,\gamma_1 \ldots \gamma_r} h_{(t_{k+1},t_k)}^{q,\xi_1 \ldots \xi_s} E\{\delta x_{k,1}^{\gamma_1} \ldots \delta x_{k,m}^{\gamma_r} \delta x_{k,1}^{\xi_1} \ldots \delta x_{k,m}^{\xi_s}\} + \\
&\quad -\delta n_{k+1}^{p} \delta n_{k+1}^{q} + R_{k+1}^{pq}, \\
P_{k+1,ip}^{xz} &= \sum_{r=1}^{v} \sum_{s=1}^{v} \frac{1}{r!s!} \Phi_{(t_{k+1},t_k)}^{i,\gamma_1 \ldots \gamma_r} h_{(t_{k+1},t_k)}^{p,\xi_1 \ldots \xi_s} E\{\delta x_{k,1}^{\gamma_1} \ldots \delta x_{k}^{\gamma_r} \delta x_{k,m}^{\xi_1} \ldots \delta x_{k}^{\xi_s}\} + \\
&\quad -\delta m_{k+1}^{i} \delta n_{k+1}^{p}, \\
\boldsymbol{K}_{k+1} &= \boldsymbol{P}_{k+1}^{xz} (\boldsymbol{P}_{k+1}^{zz})^{-1}, \\
\boldsymbol{m}_{k+1}^{+} &= \boldsymbol{m}_{k+1}^{-} + \boldsymbol{K}_{k+1}(\boldsymbol{z}_{k+1} - \boldsymbol{n}_{k+1}^{-}), \\
\boldsymbol{P}_{k+1}^{+} &= \boldsymbol{P}_{k+1}^{-} - \boldsymbol{K}_{k+1} \boldsymbol{P}_{k+1}^{zz} \boldsymbol{K}_{k+1}^{T},
\end{aligned} \tag{7}$$

where $\delta n_{k+1}^{p} = h_p(\boldsymbol{\Phi}(t_{k+1}; \boldsymbol{m}_k^{+}, t_k), t_{k+1}) - n_{k+1,p}^{-}$.

If the case of variables with Gaussian random distributions is considered, the higher-order moments $E\{\delta x_k^{\gamma_1} \ldots \delta x_k^{\gamma_p}\}$ can be completely described, at first, 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 [15]. This is an approximation of the actual propagated probability density function since, due to the nonlinear dynamics, the system loses its Gaussian assumption. Finally, it is worth stressing that, in the DA framework, the high-order partials derivation and integration, required by the standard HNEKF, is completely avoided.

IV. Unscented Kalman Filters

Differently from the HNEKF, the prediction step of the Unscented Kalman Filter (UKF) relies on the unscented transformation (UT). Such transformation is based on the intuition that *it is easier to approximate a Gaussian (probability)*

distribution than to approximate an arbitrary nonlinear function or transformation (see [16–18]). Following this statement, the aim of the unscented transformation is to find a parametrization which accurately describes the mean and the covariance information of the initial variable **x** and, at the same time, permits the direct propagation of the information through the set of nonlinear equations (e.g. functions, transformations,..). In order to approximate the mean $\boldsymbol{m}$ and covariance $\boldsymbol{P}$ of the m-dimensional state vector $\boldsymbol{x}$, a set of $2m + 1$ sigma points, collected in a structure $\mathcal{X}$, are exploited and computed as follows:

$$\mathcal{X}^{\{1\}} = \boldsymbol{m}, \tag{8}$$

$$\mathcal{X}^{\{i\}} = \boldsymbol{m} + (\sqrt{(m+\lambda)\boldsymbol{P}}\,)_{i-1}, \qquad i = 2, ..., m+1 \tag{9}$$

$$\mathcal{X}^{\{i\}} = \boldsymbol{m} - (\sqrt{(m+\lambda)\boldsymbol{P}}\,)_{i-m-1}, \qquad i = m+2, ..., 2m+1 \tag{10}$$

where λ is a scaling parameter, while $(\sqrt{(m+\lambda)\boldsymbol{P}}\,)_i$ is the ith row or column of the matrix square root of $(m+\lambda)\boldsymbol{P}$. Defining $\boldsymbol{S}$ such that $\boldsymbol{P} = \boldsymbol{SS}$, the matrix square root is computed through diagonalization:

$$\begin{aligned}
\boldsymbol{P} &= \boldsymbol{VDV}^{-1} \\
&= \boldsymbol{V}\begin{pmatrix} d_{11} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & d_{mm} \end{pmatrix}\boldsymbol{V}^{-1} \\
&= \boldsymbol{V}\begin{pmatrix} \sqrt{d_{11}} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \sqrt{d_{mm}} \end{pmatrix}\begin{pmatrix} \sqrt{d_{11}} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \sqrt{d_{mm}} \end{pmatrix}\boldsymbol{V}^{-1}.
\end{aligned} \tag{11}$$

Therefore, the matrix square root of the covariance $\boldsymbol{P}$ can be obtained as:.

$$\boldsymbol{S} = \boldsymbol{V}\begin{pmatrix} \sqrt{d_{11}} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \sqrt{d_{nn}} \end{pmatrix}\boldsymbol{V}^{-1}. \tag{12}$$

So that:

$$\boldsymbol{SS} = (\boldsymbol{VD}^{1/2}\boldsymbol{V}^{-1})(\boldsymbol{VD}^{1/2}\boldsymbol{V}^{-1}) = \boldsymbol{P}. \tag{13}$$

In this paper the Cholesky Matrix Square Root is adopted. This decomposition, often used in UKF implementation, is efficient and stable, and thus particularly suitable for real-time estimation [19] [17].

Defined the sigma points, two weights are associated to each of them, computed as in Eqs. 14-15-16.

$$w_{m,1} = \frac{\lambda}{m + \lambda}, \tag{14}$$

$$w_{P,1} = w_{m,1} + (1 - \alpha^2 + \beta), \tag{15}$$

$$w_{m,i} = w_{P,i} = \frac{1}{2(m + \lambda)}, \qquad i = 2, ..., 2m + 1 \tag{16}$$

$w_{m,i}$ is a weight referred to the mean, while $w_{P,i}$ is referred to the covariance. Parameters $k \geq 0$ and $\alpha \in (0, 1]$ define how far from the mean the sigma points are located. β is typically chosen equal to 2, since this value is the optimal choice for Gaussian distributions. Finally, λ is a combination of the other parameters:

$$\lambda = \alpha^2(m + k) - m. \tag{17}$$

It can be noted that there is no unique solution for the sigma points vector and the weights vectors, thus the sigma points can (but do not have to) lie on the main axes of the covariance matrix. However, the selection must be such that the following equations are respected:

$$\sum_{i=1}^{2m+1} w_i = 1, \tag{18}$$

$$\boldsymbol{m} = \sum_{i=1}^{2m+1} w_{m,i} \mathcal{X}^{\{i\}}, \tag{19}$$

$$\boldsymbol{P} = \sum_{i=1}^{2m+1} w_{P,i} (\mathcal{X}^{\{i\}} - \boldsymbol{m})(\mathcal{X}^{\{i\}} - \boldsymbol{m})^T. \tag{20}$$

The UKF prediction starts with a set of sigma points around the initial conditions, and their relative weights. Afterwards, the set gets transformed by initializing each sigma point through the process model, giving the transformed set (Eq. 21). Finally, the mean and the covariance of the propagated set is evaluated by weighting each transferred sigma point (Eqs. 22-23).

$$\mathcal{X}_{k+1}^{\{i\}} = \boldsymbol{\Phi}(t_{k+1}; \mathcal{X}_k^{\{i\}}, t_k), \tag{21}$$

$$\boldsymbol{m}_{k+1}^- = \sum_{i=1}^{2m+1} w_{m,i} \mathcal{X}_{k+1}^{\{i\}}, \tag{22}$$

$$\boldsymbol{P}_{k+1}^- = \sum_{i=1}^{2m+1} w_{P,i} \{\mathcal{X}_{k+1}^{\{i\}} - \boldsymbol{m}_{k+1}^-\}\{\mathcal{X}_{k+1}^{\{i\}} - \boldsymbol{m}_{k+1}^-\}^T. \tag{23}$$

The same approach is adopted for the measurement vector, where each sigma point goes through the observation model, and then the predicted measurements are calculated:

$$\mathcal{Z}_{k+1}^{\{i\}} = \boldsymbol{h}(\mathcal{X}_{k+1}^{\{i\}}, t_{k+1}), \tag{24}$$

$$\boldsymbol{n}_{k+1}^{-} = \sum_{i=1}^{2m+1} w_{m,i} \mathcal{Z}_{k+1}^{\{i\}}. \tag{25}$$

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 evaluated as:

$$\boldsymbol{P}_{k+1}^{zz} = \sum_{i=1}^{2m+1} w_{P,i} \{\mathcal{Z}_{k+1}^{\{i\}} - \boldsymbol{n}_{k+1}^{-}\} \{\mathcal{Z}_{k+1}^{\{i\}} - \boldsymbol{n}_{k+1}^{-}\}^T + \boldsymbol{R}_{k+1}, \tag{26}$$

$$\boldsymbol{P}_{k+1}^{xz} = \sum_{i=1}^{2m+1} w_{P,i} \{\mathcal{X}_{k+1}^{\{i\}} - \boldsymbol{m}_{k+1}^{-}\} \{\mathcal{Z}_{k+1}^{\{i\}} - \boldsymbol{n}_{k+1}^{-}\}^T. \tag{27}$$

This completes the UKF prediction process and the filtering can continue with the update equations of the classical Kalman filter.

A. DA-based UKF

The UKF can be developed into the DA framework becoming the DA-based Unscented Kalman Filter (UKFDA). This filter provides an improvement by reducing the computational time of the classical UKF when the equations of motion are complex. DA is used to Taylor expand the function $\boldsymbol{\Phi}$: 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, replacing multiple integrations of $\boldsymbol{\Phi}$. Consequently, the DA-based approach tends to outperform the classical one when the integration of $\boldsymbol{\Phi}$ is computationally demanding. The order at which the Taylor polynomial is computed can be arbitrarily selected.

More specifically, at each step, the state $\boldsymbol{x}$ 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 sigma point is known and given by the columns (or rows) of matrix $\boldsymbol{S} = \pm\sqrt{(m+\lambda)\boldsymbol{P}}$. The sigma points are propagated by simply evaluating the Taylor expansion at each column of $\boldsymbol{S}$. As a results, the UKFDA turns out to be faster than the plain UKF. Figure 3 gives a visual idea of the different propagation technique used by the two unscented Kalman filters.

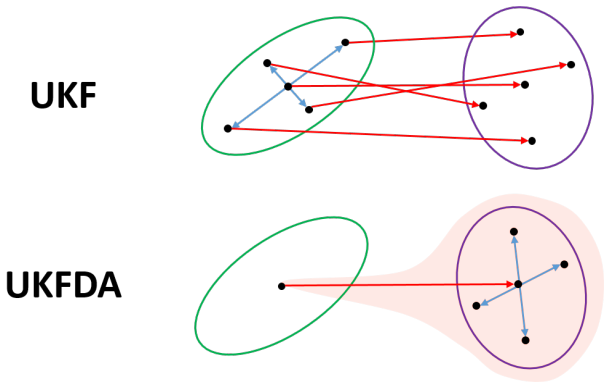

Fig. 3 The propagation approaches of the sigma points in the UKF, classic, and UKFDA, through polynomial transition map.

V. Relative Pose Estimation

This paper focuses on exploiting the proposed DA-based HNEKF and DA-based UKF to face the challenging problem of estimating the spacecraft state for proximity operations during a rendezvous with an uncooperative target. In particular, the ESA e.deorbit mission [13] is considered as reference and Envisat is selected as target satellite.

In the following analysis, some assumptions are made. Firstly, an *a priori* knowledge of both chaser and target is assumed, i.e. the inertia properties are perfectly known. Secondly, the chaser motion is supposed to be deterministic and, thus the related data are not affected by noise and uncertainties. Finally, neither flexible dynamics nor external disturbances are considered. It should be noticed that neglecting external disturbances and flexibility implies the decoupling of the relative translational and rotational dynamics.

A. Relative translational dynamics

The relative translational dynamic equations are developed in the local vertical local horizontal (LVLH) frame fixed on the chaser. In this frame the target relative position $\boldsymbol{r}_r$ and velocity $\boldsymbol{v}_r$ can be defined as:

$$\boldsymbol{r}_r = x\hat{\boldsymbol{r}} + y\hat{\boldsymbol{\theta}} + z\hat{\boldsymbol{h}}, \tag{28}$$

$$\boldsymbol{v}_r = \dot{x}\hat{\boldsymbol{r}} + \dot{y}\hat{\boldsymbol{\theta}} + \dot{z}\hat{\boldsymbol{h}}, \tag{29}$$

where x, y and z are the three components of $\boldsymbol{r}_r$ in the chaser LVLH frame and $\hat{\boldsymbol{r}}$, $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{h}}$ are the versors of the considered triad in the radial, transverse, and normal directions, respectively. The relative translational dynamics are governed by the following equations [20]:

$$\ddot{x} - 2\dot{\nu}\dot{y} - \ddot{\nu}y - \dot{\nu}^2 x = -\mu(\bar{r} + x)/[(\bar{r} + x)^2 + y^2 + z^2]^{3/2} + \mu/\bar{r}^2, \tag{30}$$

$$\ddot{y} + 2\dot{\nu}\dot{x} + \ddot{\nu}x - \dot{\nu}^2 y = -\mu y/[(\bar{r} + x)^2 + y^2 + z^2]^{3/2}, \tag{31}$$

$$\ddot{z} = -\mu z/[(\bar{r} + x)^2 + y^2 + z^2]^{3/2}, \tag{32}$$

where μ is the gravitational parameter, $\bar{r}$ is the distance from the Earth center to the chaser and ν is the true anomaly. Finally, the motion of the chaser is described by the following equations:

$$\ddot{\bar{r}} = \bar{r}\dot{\nu}^2 - \mu/\bar{r}^2, \tag{33}$$

$$\ddot{\nu} = -2\dot{\bar{r}}\dot{\nu}/\bar{r}. \tag{34}$$

B. Relative rotational dynamics

As for the rotational dynamics, the relative orientation of the body-fixed reference frame on the target with respect to the body-fixed reference frame on the chaser can be described through a rotation matrix $\mathbf{\Gamma}$. Consequently, the relative angular velocity and acceleration of the target can be expressed as follows:

$$\boldsymbol{\omega}_r = \boldsymbol{\omega}_t - \mathbf{\Gamma}\boldsymbol{\omega}_c, \tag{35}$$

$$\dot{\boldsymbol{\omega}}_r = \dot{\boldsymbol{\omega}}_t - \mathbf{\Gamma}\dot{\boldsymbol{\omega}}_c + \dot{\boldsymbol{\omega}}_{app}, \tag{36}$$

$$\dot{\boldsymbol{\omega}}_{app} = \boldsymbol{\omega}_r \times \mathbf{\Gamma}\boldsymbol{\omega}_c, \tag{37}$$

where $\boldsymbol{\omega}_c$ and $\boldsymbol{\omega}_t$ are the angular velocity of the chaser and the target expressed in their body-fixed reference frame, respectively, whereas $\boldsymbol{\omega}_r$ is the relative angular velocity expressed in the target body-fixed reference frame.

The relative attitude of the target can be described parameterizing the rotation matrix $\mathbf{\Gamma}$. To this aim, the Modified Rodrigues Parameters (MRP) are adopted in this study [21]. The MRP are related to quaternions and to the rotation matrix by the following relations:

$$\boldsymbol{\zeta} = \frac{\tilde{\boldsymbol{q}}}{1 + q_0}, \tag{38}$$

$$\boldsymbol{\Gamma}(\boldsymbol{\zeta}) = \boldsymbol{I}_3 - \alpha_1^A[\boldsymbol{\zeta}\times] + \alpha_2^A[\boldsymbol{\zeta}\times]^2, \tag{39}$$

$$\begin{cases} \alpha_1^A = 4\frac{1-\boldsymbol{\zeta}^T\boldsymbol{\zeta}}{(1+\boldsymbol{\zeta}^T\boldsymbol{\zeta})^2} \\ \alpha_2^A = 8\frac{1}{(1+\boldsymbol{\zeta}^T\boldsymbol{\zeta})^2}, \end{cases} \tag{40}$$

where $\boldsymbol{\zeta}$ are the MRP, $\tilde{\boldsymbol{q}}$ and q_0 are the vector and scalar part of the quaternions, respectively, and $\boldsymbol{I}_3$ is the identity matrix.

The time evolution of the MRP is governed by Eq. 41.

$$\dot{\boldsymbol{\zeta}} = \frac{1}{4}\boldsymbol{\Sigma}(\boldsymbol{\zeta})\boldsymbol{\omega}_r, \tag{41}$$

$$\boldsymbol{\Sigma}(\boldsymbol{\zeta}) = (1 - \boldsymbol{\zeta}^T\boldsymbol{\zeta})\boldsymbol{I}_3 + 2\boldsymbol{\zeta}\boldsymbol{\zeta}^T + 2[\boldsymbol{\zeta}\times]. \tag{42}$$

As for the dynamics, the chaser motion is described by the torque-free Euler equations, while the relative attitude dynamics can be obtained substituting kinematics relationship in the Euler absolute equations of the target spacecraft. The resulting dynamic system is:

$$\boldsymbol{J}_t\dot{\boldsymbol{\omega}}_r + \boldsymbol{\omega}_r \times \boldsymbol{J}_t\boldsymbol{\omega}_r = \boldsymbol{M}_{app} - \boldsymbol{M}_g - \boldsymbol{M}_{ci}, \tag{43}$$

$$\boldsymbol{M}_{app} = \boldsymbol{J}_t\boldsymbol{\omega}_r \times \boldsymbol{\Gamma}\boldsymbol{\omega}_c, \tag{44}$$

$$\boldsymbol{M}_{ci} = \boldsymbol{J}_t\boldsymbol{\Gamma}\dot{\boldsymbol{\omega}}_c, \tag{45}$$

$$\boldsymbol{M}_g = \boldsymbol{M}_{gc} + \boldsymbol{M}_{gcoup}, \tag{46}$$

$$\boldsymbol{M}_{gc} = \boldsymbol{\Gamma}\boldsymbol{\omega}_c \times \boldsymbol{J}_t\boldsymbol{\Gamma}\boldsymbol{\omega}_c, \tag{47}$$

$$\boldsymbol{M}_{gcoup} = (\boldsymbol{\omega}_r \times \boldsymbol{J}_t\boldsymbol{\Gamma}\boldsymbol{\omega}_c + \boldsymbol{\Gamma}\boldsymbol{\omega}_c \times \boldsymbol{J}_t\boldsymbol{\omega}_r), \tag{48}$$

where $\boldsymbol{J}_t$ is the matrix of inertia of the target, $\boldsymbol{M}_{app}$ is the apparent torques, $\boldsymbol{M}_{ci}$ is the chaser-inertial torques and $\boldsymbol{M}_g$ is

the gyroscopic torques.

C. Measurement model

In real applications, the image processing software is set up to look for some target points in each image taken: these points are referred to as markers [22]. Therefore, a measurement model that considers marker positions instead of directly states is developed. 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. A common solution is to select the target corners as markers. Assuming to have information about Envisat's mass, dimensions, center of mass (CM) location, moments of inertia, geometrical center (GC) and volume[23], markers can provide information about the spacecraft position and attitude, since their location with respect to the center of mass ($\boldsymbol{\varrho}_i$), in the target body-fixed reference frame, is well known (see Table 1).

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. Each marker is called with an alphabetical letter in order to have a clear identification: hence, there are Marker A, B, C, D, E, F, G, and H.

Marker	$\boldsymbol{\varrho}_i$ [m]		
A	+8.9150	+1.3840	+1.5970
B	+8.9150	+1.3840	-1.6030
C	+8.9150	-1.3660	-1.6030
D	+8.9150	-1.3660	+1.5970
E	-1.1050	+1.3840	+1.5970
F	-1.1050	+1.3840	-1.6030
G	-1.1050	-1.3660	-1.6030
H	-1.1050	-1.3660	+1.5970

Table 1 Assumed Envisat markers position vectors with respect to its center of mass.

Being $\boldsymbol{r}_r$ the position vector of the target center of mass with respect to chaser center of mass, the measurements are calculated separately for each single marker in the following way:

$$z_i = \boldsymbol{\Gamma}^T \boldsymbol{\varrho}_i + \boldsymbol{r}_r, \qquad i = A, \ldots, H \tag{49}$$

where z_i is the position of marker i with respect to the chaser center of mass and the rotation matrix $\boldsymbol{\Gamma}$ comes from the knowledge of the MRP. The visibility of the markers is assessed using the *prior* probability density function, where the prediction of $\boldsymbol{\Gamma}$ directly connects to the markers that are seen by the camera. Therefore, the line of sight of the markers is defined through a face visibility analysis, as shown later in the paper. Only the needed predicted measurements are evaluated as in Eq. 49. Afterwards, noise is introduced additively as an exponentially correlated random variable

according to the following model:

$$E(t_{k+1}) = KE(t_k) + \sqrt{1 - K^2} \cdot \mathcal{N}(0, \sigma), \tag{50}$$

$$K = exp(-1/(f\tau)), \tag{51}$$

where E is the error with respect to the true states, $\mathcal{N}(0, \sigma)$ is a random number generated with a normal distribution of zero mean and standard deviation σ, f is the measurement acquisition frequency and τ is the autocorrelation time. In this model, the error at time $k + 1$ is exponentially correlated to the error at the previous instant, and this correlation decays with a time scale defined by τ. Considering a camera, this seems to be a more reasonable model than the Gaussian one, in which error values at different time instant are completely uncorrelated[24].

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 structure will cover some markers. The visibility and the correct association of a corner to the correspondent marker is fundamental since larger is the number of located markers better is the estimation accuracy. Therefore, the markers visibility must be assessed.

When thinking about visibility of the corners of a parallelepiped, it is better to understand which face of the parallelepiped is visible and then associate the respective corners. Basically, there are only three different options: 1, 2 or 3 faces are visible, and thus 4, 6 or 7 markers are visible in each frame.

In order to implement this process, a set of unit vectors $\hat{\boldsymbol{\eta}}_i$, with $i = \alpha, \ldots, \zeta$ indicating the faces, is defined in the target body-fixed reference frame. In Table 2 the faces with the associated markers and unit vectors are listed.

The requirement for the face visibility is expressed by the following inequality:

$$\boldsymbol{r}_r \cdot \boldsymbol{\Gamma}^T \hat{\boldsymbol{\eta}}_i < 0. \tag{52}$$

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.

In the following, the performance of the filters will be assessed both exploiting the whole set of available markers, which means a shape-shifting measurement vector that adapts to the number of visible markers for each acquisition, and limiting the set to three markers, namely the minimum number to derive the target state.

In the latter case, the selection of the markers is based on a simple criterion: the filter has to work with the 3 markers creating the triangle with the largest area on the plane of sight. The plane of sight, from now on called π_1, is the plane defined by the relative position chaser-target vector and passing through the target center of mass. This plane gives

Face	$\hat{\eta}_i$	Markers Seen
α	$\hat{\eta}_\alpha = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$	A - B - C - D
β	$\hat{\eta}_\beta = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$	A - B - E - F
γ	$\hat{\eta}_\gamma = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$	A - D - E - H
δ	$\hat{\eta}_\delta = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}$	E - F - G - H
ϵ	$\hat{\eta}_\epsilon = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$	C - D - G - H
ζ	$\hat{\eta}_\zeta = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$	B - C - F - G

Table 2 Envisat main body faces and visible markers relations.

information on how the camera sees the target. Therefore, each marker is projected on π_1 and its projection represents the vertex of a number of triangles depending on the number of visible markers. Afterwards, the area of the triangles is evaluated and the maximum one is selected.

The marker projection is performed as follows:

$$b_i = \frac{\boldsymbol{r}_r}{||\boldsymbol{r}_r||} \cdot \boldsymbol{\Gamma}^T \boldsymbol{\varrho}_i. \tag{53}$$

$$\tilde{\boldsymbol{\varrho}}_i = \boldsymbol{\Gamma}^T \boldsymbol{\varrho}_i - b_i \frac{\boldsymbol{r}_r}{||\boldsymbol{r}_r||}, \tag{54}$$

where $\tilde{\boldsymbol{\varrho}}_i$ is the vector projection of the marker i on π_1, and b_i is the scalar projection of the marker vector $\boldsymbol{\varrho}_i$ on direction $\frac{\boldsymbol{r}_r}{||\boldsymbol{r}_r||}$.

In order to identify the markers forming the triangle with the maximum area, it is convenient to notice that Envisat geometrical model is a slender parallelepiped with one predominant principal inertia axis $\hat{\boldsymbol{\upsilon}}$, expressed here in the chaser body-fixed reference frame, and subdivide the markers into two groups at the ends of this axis: the *master* group and *slave* group. The former group presents more visible markers that the latter one. Then, it is possible to define a line l through a versor $\tilde{\boldsymbol{\upsilon}}$ obtained by projecting the axis $\hat{\boldsymbol{\upsilon}}$ on plane π_1, as reported in Eq. 55.

$$\tilde{\boldsymbol{v}} = \hat{\boldsymbol{v}} - (\frac{\boldsymbol{r}_r}{||\boldsymbol{r}_r||} \cdot \hat{\boldsymbol{v}}) \frac{\boldsymbol{r}_r}{||\boldsymbol{r}_r||}. \tag{55}$$

The markers combination that has the triangle with the largest area can be directly found by selecting the two furthest marker projections from line l in the *master* group and the nearest marker projection in the *slave* group. The distance d_i of each marker projection from line l is evaluated in the following way:

$$d_i = ||\tilde{\boldsymbol{\varrho}}_i \wedge \tilde{\boldsymbol{v}}||. \tag{56}$$

Virtually, 56 combinations of 3 markers can be identified (binomial coefficient of 8 over 3), since the total number of markers is 8. However, some combinations are physically impossible due to Envisat geometry model, i.e. opposite markers can not be visible at the same time. Hence, there are 4 pairs of markers that will never be part of the same group: AG, BH, CE and DF, and they lead to a total of 24 *impossible* combinations. Moreover, 3 markers belonging to the same side of the parallelepiped can not be chosen, and thus only 24 combination are actually allowed. Table 3 reports and classifies all the combinations of 3 markers.

	Groups							
	ABG	ACG	ADG	AEG	AFG	AGH	ABH	BCH
impossible	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
	ABE	ABF	BCF	BCG	CDH	CDG	ADE	ADH
allowed	ACF	ACH	BDE	BDG	AEF	BEF	BFG	CFG
	CGH	DGH	AEH	DEH	BEG	DEG	AFH	CFH

Table 3 Classification of all possible combination of 3 markers.

D. Software architecture

Fig. 4 reports the software architecture, which is made up of three main blocks. The first one is the "dynamics simulator+noise generator" that receives as inputs the initial states, then propagates the dynamics through a variable-step integrator (Runge-Kutta78) and generates the measurements adding noise computed with the exponentially correlated random model. These computations are performed in advance and the outputs are loaded in memory before running the filter.

For the filtering, the decoupling of the equations of motion can be exploited to separate the propagation of the two dynamics, translational and rotational, leading to a faster and more efficient algorithm. In this case a fixed-step integrator is used, because computationally lighter. On the other hand, as regards the estimation of the measurements, the translational and rotational information has to be used jointly, since the measurement equations are coupled.

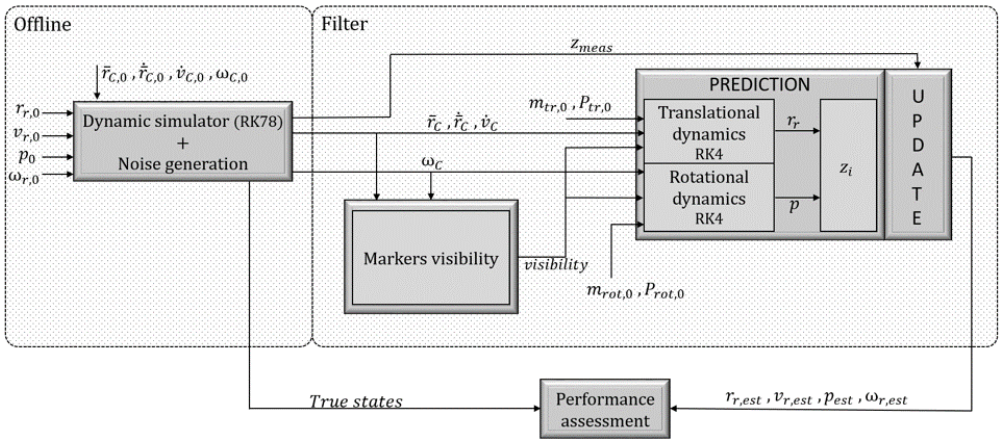

Fig. 4 Software architecture.

The filter is initialized providing an initial estimate of the relative states in terms of mean and covariance. Moreover, before starting the estimation, the filter uses the information of the previous step to calculate the markers visibility.

Finally, the estimated relative state is compared with the true state propagated by the dynamics simulator to assess the performance of the filters.

VI. Results

In the numerical analysis, the chaser and the target are assumed to be on the same orbit at a reasonable distance for a proper functioning of the camera. The initial conditions of the relative states are reported in Table 4. The attitude is initialized randomly, while the angular velocity is selected in order to have an absolute value of about 2.5 deg/s.

The assessment of the performance of the DA-based HNEKF is limited to the use of first and second order expansions in the following analyses. Indeed, no relevant improvement is obtained with higher orders, since Kalman filters are based on a Gaussian representation of the propagated uncertainties[12].

Tr. Dyn.		Rot. Dyn.	
x (m)	-0.002	ϕ (rad)	1.66
y (m)	-31.17	θ (rad)	2.27
z (m)	0	ψ (rad)	-0.38
$\dot{x}$ (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 4 Initial conditions.

Dynamics		Sensors		Frequency
$\sigma_{r_r,0}$ (m)	K*1	$\sigma^s_{x,y}$ (m)	0.02	0.05 Hz to 3 Hz
$\sigma_{v_r,0}$ (m/s)	K*0.1	σ^s_z (m)	0.03	
$\sigma_{\zeta,0}$ ()	K*0.002			0.05 Hz to 3 Hz
$\sigma_{\omega_r,0}$ (rad/s)	K*0.01			
K = [1,5,10]				

Table 5 Sensitivity to acquisition frequency, initial velocity and angular velocity uncertainty.

A. Accuracy and robustness analysis

Before presenting the results, some comments are provided to guide the reader through the following analyses. First, the target velocity can be assumed to be the most uncertain variables since neither *a priori* knowledge nor direct measurements are available. Then, when limited-resource systems are considered, low measurement acquisition frequency could be imposed or at least beneficial. Therefore, a Monte Carlo-based sensitivity analysis is carried out to assess the robustness of first and second order filters with various acquisition frequencies and initial uncertainty reported in Table 5, with $\sigma_{i,0}$ and σ^s_i being the initial standard deviation and the sensor standard deviation, respectively, of the variable i.

For each case, 1000 samples are generated around the true initial conditions, according to the statistics, and then the furthest 100 are selected and used as initial estimate of the relative state in the filter. This choice is motivated by the will to study the worst circumstances, in which the nonlinearities are expected to play a prominent role.

Afterwards, the performance are quantified by means of some statistical indexes, reported in Eqs. (57) and (58).

$$_{filter}\bar{\mu} = \frac{\sum_{i=1}^{100} RMSE_i}{100} \tag{57}$$

$$_{filter}\sigma_{\bar{\mu}} = \left[\frac{\sum_{i=1}^{100}(_{filter}\bar{\mu} - RMSE_i)^2}{100}\right]^{\frac{1}{2}} \tag{58}$$

$RMSE_i$ is the root mean square error of the estimated variables computed at steady state for the i^{th} simulation, $_{filter}\bar{\mu}$ and $_{filter}\sigma_{\bar{\mu}}$ are the mean and the standard deviation of RMSE, respectively. Fig. 5 provides a deeper insight of the indexes: $\bar{\mu}$ gives the mean accuracy of the filter, while $\sigma_{\bar{\mu}}$ quantifies the dispersion around the mean. If the standard deviation is high, the final accuracy strongly depends on the estimate of the initial conditions and thus large initial errors may result in bad performance or even failure. The statistical indexes are computed considering only the simulations that converge reaching a final error at steady state one order of magnitude lower than the initial error.

In the following, the case with 3 markers is deeply analyzed because it is the most significant. The results and conclusions considering the case with the whole set of markers are basically the same, since the extended set just slightly

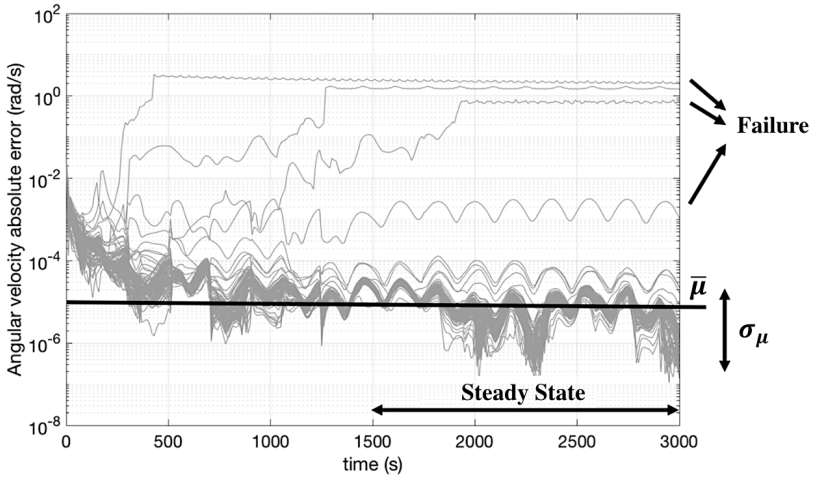

Fig. 5 Graphical representation of the statistical indexes.

improves the accuracy of the filters.

The translational dynamics is almost static and linear since the two spacecraft are very close on the same orbit, which is nearly circular. In [12], it is shown that high-order filters do not provide better performance with respect to the linear one, which is already capable of following the dynamic evolution. However, differently from [12], here the measurement equations are nonlinear and coupled. In particular, the performance of the translational filter strictly depends on the performance of the rotational one. Indeed, the rotational states are the most demanding ones to be estimated, due to the high nonlinearities in the dynamics. At the higher acquisition frequencies, both first and second order filters provide good performance with the similar error at steady state, which is in the order of 10^{-4} m for the position and 10^{-7} m/s for the velocity. The estimation of the translational states deteriorates at lower frequency due to the coupling with the rotational states. Consequently, the rotational filter is deemed to deserve a deeper study, which is reported in the following. In Fact, as already stated, the nonlinearities affect the estimation more significantly, especially in case of high uncertainties and low observability of the system, and thus it merits a more detailed analysis.

From now on, the first and second order DA-based HNEKF are referred to as EKFDA1 and EKFDA2, respectively, while the DA-based UKF is referred to as UKFDA2 since it exploits a second order Taylor expansion of the flow.

From Tables 6-7, it can be noticed that filters do not show significant difference in the performance for low uncertainties and high acquisition frequency. However, the situation changes moving to high uncertainties and low frequency. Especially, it can be observed how the convergence was not achieved with the EKFDA (both 1 and 2) in the case with frequency 0.05 Hz and $\mathbf{K} = 10$ and fewer samples reached convergence for similar scenarios. The UT-based filters have higher convergence rate of samples at low frequency with respect to DA-based HNEKF filters, but the EKFDA2 still shows slightly better accuracy. On the other hand, the EKFDA1 is the least accurate filter. Indeed, even if the improvement gained by high order filters in terms of the mean error accuracy is not so relevant, ${}_{EKFDA1}\sigma_{\bar{\mu}}$ shows

Modified Rodrigues Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
$EKFDA1\bar{\mu}$							
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
$EKFDA1\sigma_{\bar{\mu}}$							
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
$EKFDA2\bar{\mu}$							
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
$EKFDA2\sigma_{\bar{\mu}}$							
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
$UKFDA2\bar{\mu}$							
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
$UKFDA2\sigma_{\bar{\mu}}$							
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 6 3 markers: sensitivity analysis for rotational dynamics.

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

Table 7 3 markers: success rate for each combination of frequency and amplitude factor, for rotational dynamics.

marked higher values, namely an higher dispersion of the final error. This means that, in case of large deviations from the true initial conditions, the first order filter performance deteriorates leading to final estimates worse than the second order filters. To visualize this result, see Fig. 6. Note that in Fig. 6 all the simulations that converge are reported to highlight the concept.

Hence, it is clear that filters that work with the full knowledge of the first two moments perform better with respect to the filter based on linearization. Among them, EKFDA2 seems to appear the filter with the best accuracy, especially for low frequencies and high uncertainties, indeed it presents lower standard deviations.

The filters based on the UT, i.e. UKF and UKFDA2, provide the same results: their difference is not at the accuracy level but on the computational time required by the filter itself, as it will be shown later. In fact, the introduction of the DA inside the UKF has the purpose of implementing a faster filter, while the accuracy is not improved.

For completeness, Table 8 reports the results considering the whole set of visible markers; as already explained, the conclusions that can be drawn are the same of the 3-markers case.

B. Computational time

Table 9 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 is evaluated as a mean among all the simulations in which the error converges. The computational mean time is evaluated in the following way:

$$\Psi_f = \frac{\sum_{j=1}^{3} \frac{\sum_{i=1}^{\kappa_{f,j}} \tau_{f,j,i}}{\kappa_{f,j}}}{3}, \tag{59}$$

where Ψ_f is the computational mean time associated to frequency $f = [0.05, 0.1, 0.5, 1, 3]\,\text{Hz}$; 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 simulations in which convergence is achieved 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 graph in Fig. 7 (left) is a graphical representation of the computational

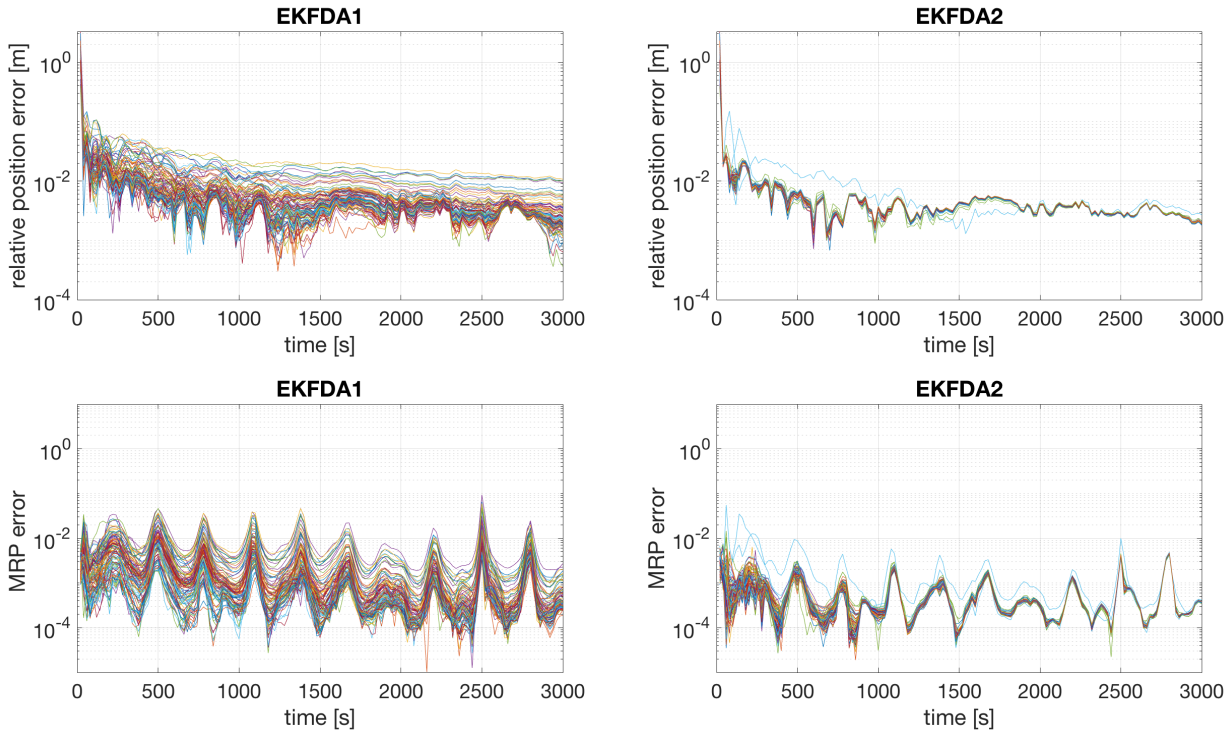

Fig. 6 EKFDA1 and EKFDA2 accuracy for the whole set of 100 samples in the case with frequency 0.05 Hz and K = 1 for the relative position and MRP.

time behavior of the filters at different frequencies in the case where all the visible markers are used as measurement. The computational time is the running-time on an Intel Core i7-6700K processor with a total of 4 cores @ 4.0 GHz and 16 GB of RAM.

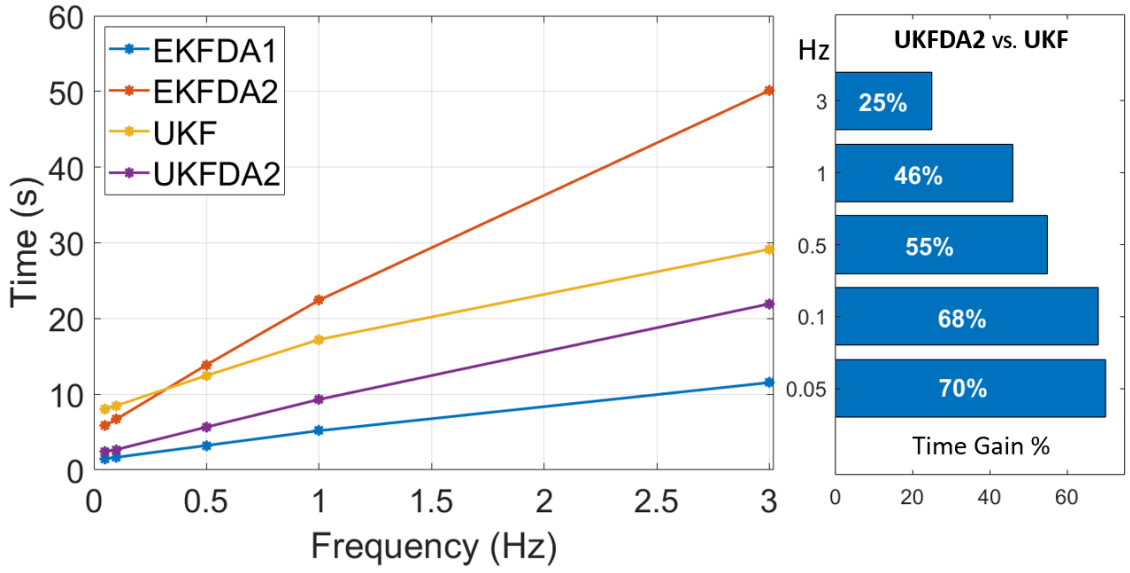

Fig. 7 All visible markers: mean computational time of the filters at different frequencies (left) and UKFDA2 prediction step time saving percentage over UKF (right).

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
$EKFDA1\bar{\mu}$							
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
$EKFDA1\sigma_{\bar{\mu}}$							
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
$EKFDA2\bar{\mu}$							
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
$EKFDA2\sigma_{\bar{\mu}}$							
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 8 All visible markers: sensitivity analysis for rotational performance.

The linear filter EKFDA1, as expected, is the fastest one. The EKFDA2 is the filter with the most demanding computational time at high frequency, while it is faster than standard UKF at very low frequency. The UKFDA2 has nearly the same trend of the UKF but it requires less time. 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 histogram in Fig. 7 (right) emphasizes the computational advantage gained by the DA implementation of the UKF. Each bar represents the percentage of workload saved by UKFDA2 (over the UKF) during all the prediction steps, for each simulation at different acquisition frequencies.

Looking at Table 9, the overall simulation time has decreased, for each filter and for all the frequencies, in the 3 markers limitation cases. The EKFDA1 is always the fastest filter, followed by UKFDA2 with almost double computational time. The main difference between the all-markers simulation and the 3-markers one lies in the comparison between EKFDA2 and UKF. The reduced length of the measurement vector implies a faster inversion of the

Freq. (Hz)	EKFDA1 Ψ_f (s)	EKFDA2 Ψ_f (s)	UKF Ψ_f (s)	UKFDA2 Ψ_f (s)
All Markers				
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
3 Markers				
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 9 Computational time analysis of the filters.

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 of time, at lower frequencies.

C. Acquisition Failure

Measurement failures can be critical in sequential state estimation application; indeed, the lack of proper data management could lead to an unforeseen behavior of the filters. Therefore, in this section, the filters are tested against marker position data failure.

Considering a set of 3 markers, 8 different situations can arise, depending on the number of markers not recognized. Indeed, using a binary notation where 1 identifies acquisition failure and 0 identifies the correct acquisition, the 8 combinations of failures are 111 - 100 - 010 - 001 - 110 - 101 - 011 - 000. Given a failure probability level, $p = 20\%$, it is possible to calculate the probability of having a certain amount of failures in the set of 3 markers. Defining with α the number of markers not recognized, then the probability of having α misrecognition in a set, $\mathcal{P}_\alpha$, is evaluated according to the following equation:

$$\mathcal{P}_\alpha = [p^\alpha (1-p)^{(N-\alpha)}]\gamma, \tag{60}$$

where N, number of elements, is in this case 3 since the set includes 3 markers, and γ is an integer number that indicates how many combination of failures having α markers not recognized are possible.

In table 10, it can be seen that only half of the time steps work without any failure (with probability threshold p),

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

Table 10 Probability of failures in a set of 3 markers.

and that the failure set of probability $\mathcal{P}_\alpha$ follows Eq. 61.

$$\sum_{\alpha=0}^{N} \mathcal{P}_\alpha = 1 \tag{61}$$

Modified Rodriguez Parameters				Relative Angular Velocity (rad/s)			
Freq.	**K**			Freq.	**K**		
(Hz)	**1**	**5**	**10**	(Hz)	**1**	**5**	**10**
$EKFDA2\bar{\mu}$							
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
$EKFDA2\sigma_{\bar{\mu}}$							
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 11 Sensitivity analysis for rotational dynamics with failure.

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

Table 12 Success rate for each combination of frequency and amplitude factor, considering failures, for rotational dynamics.

Table 11 shows that, as expected, failures affect negatively the filters performance. Robustness to acquisition failures has been tested at low frequency, where the lack of measurement becomes more relevant. Only the results of the EKFDA2 are reported to give a comparison with respect to the case without failures. It can be observed that the accuracy decreases in terms of both the mean value and standard deviation of the RMSE. However, the main difference is in the latter one. In fact, 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 12 it can be noticed that the success rate of all the filters decreases further at low frequency.

The performance comparison among the filters presents the same behavior 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. Indeed, the EKFDA1 achieves convergence, but it tends to show anomalous behaviors, meaning less robustness to failures.

VII. Conclusion

In this paper different nonlinear filtering techniques based on DA have been proposed with the aim of estimating, onboard, 6DoF state. The problem of real-time relative pose estimation during proximity operations has been considered as target application, using the e.deorbit mission with the target Envisat as reference scenario. In particular, a DA-based HNEKF and DA-based UKF have been developed, analyzed and compared. The results show that second order filters guarantee higher level of robustness and accuracy with respect to first order one, especially for relatively large initial errors and uncertainties, or for relatively low acquisition frequencies. Nevertheless, the second order filters outperform the first order filter in terms of final error dispersion and robustness to failure. Finally, it has been proven that the implementation of the UKF in the DA framework, limited to the 2^{nd} order, reduces the computational weight, providing a markedly faster algorithm. Moreover, the UKFDA2 turns out to be almost as accurate as the EKFDA2, but computationally lighter.

References

[1] Wertz, J., and Bell, R., "Autonomous rendezvous and docking technologies: status and prospects," *Proceedings of the SPIE*, Vol. 5088, 2003, pp. 20–30. Doi: 10.1117/12.498121.

[2] Montenbruck, O., and Eberhard, G., *Satellite orbits: models, methods, and applications*, Springer Verlag, 2000.

[3] Julier, S., Uhlmann, J., and Durrant-Whyte, H., "A new approach for filtering nonlinear systems," *Proceedings of the IEEE*, Vol. 3, 1995, pp. 1628–1632. Doi: 10.1109/ACC.1995.529783.

[4] Julier, S., and Uhlmann, J., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, 2004, pp. 401–422. Doi: 10.1109/JPROC.2003.823141.

[5] Park, R., and Scheeres, D., "Nonlinear mapping of Gaussian statistics: theory and applications to spacecraft trajectory design," *Journal of Guidance Control and Dynamics*, Vol. 29, No. 6, 2006, pp. 1367–1375. Doi: 10.2514/1.20177.

[6] Park, R., and Scheeres, D., "Nonlinear semi-analytic methods for trajectory estimation," *Journal of Guidance Control and Dynamics*, Vol. 30, No. 6, 2007, pp. 1668–1676. Doi: 10.2514/1.29106.

[7] Majji, M., Junkins, J., and Turner, J., "A High Order Method for Estimation of Dynamic Systems," *The Journal of the Astronautical Sciences*, Vol. 56, No. 3, 2008, pp. 401–440. Doi: 10.1007/BF03256560.

[8] Berz, M., "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*, Vol. 258, No. 3, 1987, pp. 431–436. Doi: 10.1016/0168-9002(87)90927-2.

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

[10] Valli, M., Armellin, R., Di Lizia, P., and Lavagna, M., "Nonlinear Filtering Methods for Spacecraft Navigation Based on Differential Algebra," *Acta Astronautica*, Vol. 94, 2014, pp. 363–374. Doi: 10.1016/j.actaastro.2013.03.009.

[11] Cavenago, F., Di Lizia, P., Massari, M., and Wittig, A., "On-board DA-based state estimation algorithm for spacecraft relative navigation," *7th European Conference for Aerospace Sciences (EUCASS 2017)*, 2017, pp. 1–14. Doi: 10.13009/EUCASS2017-607.

[12] Cavenago, F., Di Lizia, P., Massari, M., and Wittig, A., "On-board spacecraft relative pose estimation with high-order extended Kalman filter," *Acta Astronautica*, Vol. 158, 2019, pp. 55–67. Doi: 10.1016/j.actaastro.2018.11.020.

[13] Biesbroek, R., "The e.Deorbit ESA CDF study," *Proceedings of the SPIE*, Vol. 5088, 2003, pp. 20–30.

[14] Massari, M., Di Lizia, P., and Rasotto, M., "Nonlinear Uncertainty Propagation in Astrodynamics Using Differential Algebra and Graphics Processing Units," *Journal of Aerospace Information Systems*, Vol. 14, No. 9, 2017, pp. 493–503. Doi: 10.2514/1.I010535.

[15] Isserlis, L., "On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables," *Biometrika*, Vol. 12, 1918. Doi: 10.2307/2331932.

[16] Julier, S. J., and Uhlmann, J. K., "A new extension of the Kalman filter to nonlinear systems," *Int. symp. aerospace/defense sensing, simul. and controls*, Vol. 3, Orlando, FL, 1997, pp. 182–193. Doi: 10.1117/12.280797.

[17] Julier, S. J., and Uhlmann, J. K., "A general method for approximating nonlinear transformations of probability distributions," Tech. rep., Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, 1996.

[18] Julier, S. J., and Uhlmann, J. K., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, 2004, pp. 401–422. Doi: 10.1109/JPROC.2003.823141.

[19] Stachniss, C., "Robot Mapping - Unscented Kalman Filter," Uni Freiburg - ASI: Autonomous Intelligence Systems, ????

[20] Schaub, H., and Junkins, J. L., *Analytical mechanics of space systems*, AIAA, 2003.

[21] Marandi, S., and Modi, V., "A preferred coordinate system and the associated orientation representation in attitude dynamics," *Acta Astronautica*, Vol. 15, No. 11, 1987, pp. 833–843. Doi: 10.1016/0094-5765(87)90038-5.

[22] Wang, H., and Brady, M., "Real-time corner detection algorithm for motion estimation," *Image and vision computing*, Vol. 13, No. 9, 1995, pp. 695–703. Doi: 10.1016/0262-8856(95)98864-P.

[23] "European Space Agency, Annex C: Envisat Specification," 17/05/2016, pp. 22–23.

[24] Gil-Fernández, J., Prieto-Llanos, T., Panzeca, R., and Drai, R., "Autonomous gnc algorithms for neo impactor missions," *Proceedings of the 20th International Symposium on Space Flight Dynamics*, 2007, pp. 1–14.