M. Valli, R. Armellin, P. Di Lizia, M. Lavagna
Nonlinear Filtering Methods for Spacecraft Navigation Based on Differential Algebra
Acta Astronautica, Vol. 94, N. 1, 2014, p. 363-374
doi:10.1016/j.actaastro.2013.03.009

Nonlinear filtering methods for spacecraft navigation based on differential algebra

Monica Valli[a], Roberto Armellin[a], Pierluigi di Lizia[a], Michèle R. Lavagna[a]

[a] *Politecnico di Milano, Dipartimento di Ingegneria Aerospaziale, Via la Masa 34, 20156 Milano, Italy*

Abstract

The paper investigates the problem of nonlinear filtering applied to spacecraft navigation. Differential algebraic (DA) techniques are proposed as a valuable tool to implement the higher-order numerical and analytic extended Kalman filters. Working in the DA framework allows us to consistently reduce the required computational effort without losing accuracy. The performance of the proposed filters is assessed on different orbit determination problems with realistic orbit uncertainties. The case of nonlinear measurements is also considered. Numerical simulations show the good performance of the filter in case of both complex dynamics and highly nonlinear measurement problems.

Keywords: Nonlinear filtering, Orbit determination, Space navigation, Differential algebra

1. Introduction

The nonlinear filtering problem plays an important role in various space-related applications and especially in orbit determination and spacecraft navigation problems. Near future sample and return missions from small bodies, landing missions to the Moon, Mars and outer planets as well as interplanetary exploration missions demand navigation systems based on accurate filtering techniques that are able to perform accurate trajectory estimation in a very reduced lapse of time.

At the present time the extended Kalman filter[1,2] (EKF) is mainly used for trajectory estimation. 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. In some cases,

however, the linear assumption fails to provide an accurate realization of the local trajectory motion due to the low frequency of the estimation process as well as the nature or the limited number of measurements. In such cases, a different method that accounts for the system nonlinearity must be used. An alternative approach is the unscented Kalman filter (UKF)[3,4] that yields superior performance with respect to the EKF in highly nonlinear situations because it is based on the unscented transformation, which does not contain any linearization. Even if the asymptotic complexity of the UKF algorithm is the same as for the EKF, in practice, the UKF is often slightly slower than the EKF. In 2007 Park and Scheeres[5,6] developed two nonlinear filters - the higher-order numerical extended Kalman filter (HNEKF) and the higher-order analytic extended Kalman filter (HAEKF) - 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, but the need to derive the so-called higher-order tensors makes them in many cases - especially for a sophisticated, high fidelity system model - difficult to use due to computational complexity. Due to this critical problem, up to now the HNEKF and the HAEKF have mainly been applied to the case of linear measurements. Up to now limited work has been done to automate and speed up the derivation of the state transition tensors.[7,8]

Differential algebraic (DA) techniques are here proposed as a valuable tool to implement the HNEKF and the HAEKF, 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.[9,10,11,12] 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 arbitrarily 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 HNEKF and the DA-based HAEKF presented in this paper, both the propagation of the mean trajectory and the measurement function evaluation are carried out in the DA framework. The obtained solution map not only provides the pointwise values for the propagated state and measurements, but also provides the higher-order partials of the solution flow and of the measurement equation. This eliminates the need to calculate the higher-order tensors at each time step by solving a complex system of augmented ODE.

The proposed filters are tested on different orbit determination problems. First of all, a Sun-Earth halo orbit around the L1 point is considered to demonstrate the precise correspondence between our results and those obtained by Park and Scheeres with the original form of the HNEKF and HAEKF. Moreover, the case of an Earth orbiting satellite with realistic orbit uncertainties and nonlinear measurements is presented. Higher orders can improve the accuracy of the state determination since they can extract, from the available nonlinear measurements, more accurate information about the state of the vehicle than low order filters. Hence, numerical simulations show good performance of the filter in case of both complex dynamics and highly nonlinear measurement problems.

The paper is organized as follows. First an introduction to differential algebra and some hints on how to obtain high order expansion of the flow are presented. Then, after a brief overview about the higher-order extended Kalman filters, differential algebra is used to improve the performance of higher-order filters with respect to the original theory. Finally, the effectiveness of the method is demonstrated through numerical examples.

2. Notes on Differential Algebra

DA techniques, exploited here to obtain m-th order Taylor expansions of the flow of a set of ODE's with respect to initial condition, were devised to attempt solving analytical problems through an algebraic approach.[10] Historically, the 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 rely 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 computer environment in a similar manner as the treatment of real numbers. Referring to Fig. 1, consider two

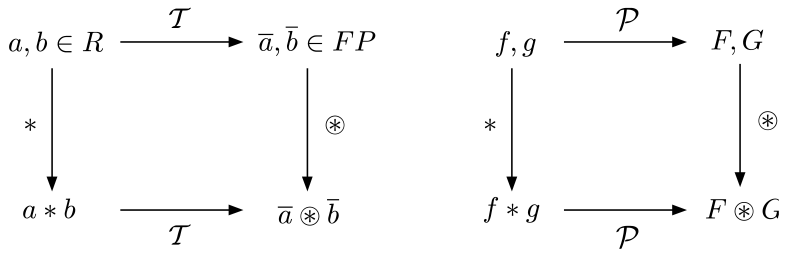


Figure 1: Analogy between the floating point representation of real numbers in a computer environment (left figure) and the introduction of the algebra of Taylor polynomials in the differential algebraic framework (right figure).

real numbers a and b. Their transformation into the floating point representation, $\overline{a}$ and $\overline{b}$ respectively, is performed to operate on them in a computer environment. Then, given any operation $*$ in the set of real numbers, an adjoint operation $\circledast$ is defined in the set of floating point (FP) numbers so that the diagram in Fig. 1 commutes. (The diagram commutes approximately in practice due to truncation errors.) Consequently, transforming the real numbers a and b into their FP representation and operating on them in the set of FP numbers returns the same result as carrying out the operation in the set of real numbers and then transforming the achieved result in its FP representation. In a similar way, let us suppose two m differentiable functions f and g in n variables are given. In the framework of differential algebra, the computer operates on them using their m-th order Taylor expansions, F and G respectively. Therefore, the transformation of real numbers in their FP representation is now substituted by the extraction of the m-th order Taylor expansions of f and g. For each operation in the space of m differentiable functions, an adjoint operation in the space of Taylor polynomials is defined so that the corresponding diagram commutes; i.e., extracting the Taylor expansions of f and g and operating on them in the space of Taylor polynomials (labeled as ${}_mD_n$) returns the same result as operating on f and g in the original space and then extracting the Taylor expansion of the resulting function. The straightforward implementation of differential algebra in a computer allows computation of the Taylor coefficients of a function up to a specified order m, along with the function evaluation, with a fixed amount of effort. The Taylor coefficients of order n for sums and products 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 func-

tions can be endowed with well-defined operations, leading to the so-called truncated power series algebra.[13,14] Similarly to the algorithms for floating point arithmetic, the algorithms for functions followed, including methods to perform composition of functions, to invert them, to solve nonlinear systems explicitly, and to treat common elementary functions.[9,10] In addition to these algebraic operations, the DA framework is endowed with differentiation and integration operators, therefore finalizing the definition of the DA structure.

2.1. The Minimal Differential Algebra

Consider all ordered pairs (q_0, q_1), with q_0 and q_1 real numbers. Define addition, scalar multiplication, and vector multiplication as follows:

$$\begin{aligned} (q_0, q_1) + (r_0, r_1) &= (q_0 + r_0, q_1 + r_1) \\ t \cdot (q_0, q_1) &= (t \cdot q_0, t \cdot q_1) \\ (q_0, q_1) \cdot (r_0, r_1) &= (q_0 \cdot r_0, q_0 \cdot r_1 + q_1 \cdot r_0). \end{aligned} \tag{1}$$

The ordered pairs with the above arithmetic are called ${}_1D_1$. The multiplication of vectors is seen to have $(1, 0)$ as the unity element. The multiplication is commutative, associative, and distributive with respect to addition. Together, the three operations defined in Eq. (1) form an algebra. Furthermore, they 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 are included.

The multiplicative inverse of the pair (q_0, q_1) in ${}_1D_1$ is

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

which is defined for any $q_0 \neq 0$.

One important property of this algebra is that it has 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

$$\begin{aligned} &(q_0, q_1) < (r_0, r_1) \quad \text{if} \quad q_0 < r_0 \quad \text{or} \quad (q_0 = r_0 \text{ and } q_1 < r_1) \\ &(q_0, q_1) > (r_0, r_1) \quad \text{if} \quad (r_0, r_1) < (q_0, q_1) \\ &(q_0, q_1) = (r_0, r_1) \quad \text{if} \quad q_0 = r_0 \text{ and } q_1 = r_1. \end{aligned} \tag{3}$$

As for any two elements (q_0, q_1) and (r_0, r_1) only one of the three relation holds, ${}_1D_1$ is said 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 $(0, 0) < (0, 1) < (r, 0) = r$. For this reason d is called an infinitesimal or a differential. In fact, d is so small that its square vanishes. Since for any $(q_0, q_1) \in {}_1D_1$

$$(q_0, q_1) = (q_0, 0) + (0, q_1) = q_0 + d \cdot q_1, \tag{4}$$

the first component is called the real part and the second component the differential part.

The algebra in ${}_1D_1$ becomes a differential algebra by introducing a map ∂ from ${}_1D_1$ to itself, and proving that the map is a derivation. Define $\partial : {}_1D_1 \to {}_1D_1$ by

$$\partial(q_0, q_1) = (0, q_1). \tag{5}$$

Note that

$$\begin{aligned} \partial\{(q_0, q_1) + (r_0, r_1)\} &= \partial(q_0 + r_0, q_1 + r_1) = (0, q_1 + r_1) \\ &= (0, q_1) + (0, r_1) = \partial(q_0, q_1) + \partial(r_0, r_1) \end{aligned} \tag{6}$$

and

$$\begin{aligned} \partial\{(q_0, q_1) \cdot (r_0, r_1)\} &= \partial(q_0 \cdot r_0, q_0 \cdot r_1 + r_0 \cdot q_1) = (0, q_0 \cdot r_1 + r_0 \cdot q_1) \\ &= (0, q_1) \cdot (r_0, r_1) + (0, r_1) \cdot (q_0, q_1) \\ &= \partial\{(q_0, q_1)\} \cdot (r_0, r_1) + (q_0, q_1) \cdot \partial\{(r_0, r_1)\} \end{aligned} \tag{7}$$

This holds for all $(q_0, q_1), (r_0, r_1) \in {}_1D_1$. Therefore ∂ is a derivation and $({}_1D_1, \partial)$ is a differential algebra.

The most important aspect of ${}_1D_1$ is that it allows the automatic computation of derivatives. Assume to have two functions f and g and to put their values and their derivatives at the origin in the form $(f(0), f'(0))$ and $(g(0), g'(0))$ as two vectors in ${}_1D_1$. If the derivative of the product $f \cdot g$ is of interest, it has just to be looked at the second component of the product $(f(0), f'(0)) \cdot (g(0), g'(0))$; whereas the first component gives the value of the product of the 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

$$[f] = (f(0), f'(0)), \tag{8}$$

it holds

$$\begin{aligned} [f+g] &= [f]+[g] \\ [f\cdot g] &= [f]\cdot[g] \end{aligned} \tag{9}$$

and

$$[1/g] = [1]/[g] = 1/[g] \tag{10}$$

by using (2). This observation can be used to compute derivatives of many kinds of functions algebraically by merely applying arithmetic rules on ${}_1D_1$, beginning from the value and the derivative of the identity function $[x] = (x, 1) = x + \delta x$. Consider the example

$$f(x) = \frac{1}{x + (1/x)} \tag{11}$$

and its derivative

$$f'(x) = \frac{(1/x^2) - 1}{(x + (1/x))^2}. \tag{12}$$

The function value and its derivative at the point $x = 3$ are

$$f(3) = \frac{3}{10}, \quad f'(3) = -\frac{2}{25}. \tag{13}$$

Evaluating the function (11) in the DA framework at $(3, 1) = 3 + \delta x$ yields

$$\begin{aligned} f((3,1)) &= \frac{1}{(3,1) + 1/(3,1)} = \frac{1}{(3,1) + (1/3, -1/9)} \\ &= \frac{1}{(10/3, 8/9)} = \left(\frac{3}{10}, -\frac{8}{9} \Big/ \frac{100}{9}\right) = \left(\frac{3}{10}, -\frac{2}{25}\right). \end{aligned} \tag{14}$$

Thus, 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 (9) and (10)

$$\begin{aligned}[f(x)] &= \left[\frac{1}{x+1/x}\right] = \frac{1}{[x+1/x]} \\ &= \frac{1}{[x]+[1/x]} = \frac{1}{[x]+1/[x]} \\ &= f([x]).\end{aligned} \tag{15}$$

2.2. The Differential Algebra ${}_mD_n$

The algebra described in this section was introduced to compute the derivatives up to an order m of functions in n variables. Similarly as before, it is based on taking the space $\mathcal{C}^m(\mathbb{R}^n)$, the collections of m times continuously differentiable functions on $\mathbb{R}^n$. On this space an equivalence relation is introduced. For f and $g \in \mathcal{C}^m(\mathbb{R}^n)$, $f =_m g$ if and only if $f(0) = g(0)$ and all the partial derivatives of f and g agree at 0 up to order m. The relation $=_m$ satisfies

$$\begin{aligned} f =_m f \quad &\text{for all} \quad f \in \mathcal{C}^m(\mathbb{R}^n), \\ f =_m g \Rightarrow g =_m f \quad &\text{for all} \quad f, g \in \mathcal{C}^m(\mathbb{R}^n), \quad \text{and} \\ f =_m g \text{ and } g =_m h \Rightarrow f =_m h \quad &\text{for all} \quad f, g, h, \in \mathcal{C}^m(\mathbb{R}^n). \end{aligned} \tag{16}$$

Thus, $=_m$ is an equivalence relation. All the elements that are related to f can be grouped together in one set, the equivalence class $[f]$ of the function 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 in all n variables up to order m. This class is called ${}_mD_n$.

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

$$\begin{aligned} [f] + [g] &= [f + g] \\ t \cdot [g] &= [t \cdot f] \\ [f] \cdot [g] &= [f \cdot g] \end{aligned} \tag{17}$$

Under this operations, ${}_mD_n$ becomes an algebra. For each $v \in 1, \ldots, n$, define the map ∂_v from ${}_mD_n$ to ${}_mD_n$ for f via

$$\partial_v[f] = \left[p_v \cdot \frac{\partial f}{\partial x_v}\right], \tag{18}$$

where

$$p_v(x_1, \ldots, x_n) = x_v \tag{19}$$

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

$$\partial_v([f] + [g]) = \partial_v[f] + \partial_v[g] \tag{20}$$

$$\partial_v([f] \cdot [g]) = [f] \cdot (\partial_v[g]) + (\partial_v[f]) \cdot [g]. \tag{21}$$

Therefore, ∂_v is a derivation for all v, and hence $({}_mD_n, \partial_1, \ldots, \partial_v)$ is a differential algebra.

Observe that f lies in the same class as its Taylor polynomial T_f of order m around the origin; they have the same function values and derivatives up to order n. Therefore,

$$[f] = [T_f] \tag{22}$$

and the introduced differential algebra is referred to as Taylor polynomial algebra.

Similar to the structure ${}_1D_1$, composition of functions and elementary functions, i.e. exp, sin, and log, are introduced in ${}_mD_n$. Consequently, the derivatives of any function f belonging to $\mathcal{C}^m(\mathbb{R}^n)$ can be computed up to order m in fixed amount of effort. The DA sketched in this section was implemented by M. Berz and K. Makino in the software COSY INFINITY.[11]

2.3. High Order Expansion of the Flow

The differential algebra allows the derivatives of any function f of n variables to be computed up to an arbitrary order m, along with the function evaluation. This has an important consequence when the numerical integration of an 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, carrying out all the evaluations in the DA framework allows differential algebra to compute the arbitrary order expansion of the flow of a general ODE with respect to the initial condition.

Without loss of generality, consider the scalar initial value problem

$$\begin{cases} \dot{x} = f(x, t) \\ x(t_0) = x_0 \end{cases} \tag{23}$$

and the associated phase flow $\varphi(t; x_0)$. We now want to show that, starting from the DA representation of the initial condition x_0, differential algebra allows us to propagate the Taylor expansion of the flow in x_0 forward in time, up to the final time t_f.

Replace the point initial condition x_0 by the DA representative of its identity function up to order m, which is a $(m+1)$-tuple of Taylor coefficients. (Note that x_0 is the flow evaluated at the initial time; i.e, $x_0 = \varphi(t_0; x_0)$.) As for the identity function only the first two coefficients, corresponding to the constant part and the first derivative respectively, are non zeros, we can write $[x_0]$ as $x_0 + \delta x_0$, in which x_0 is the reference point for the expansion. If all the operations of the numerical integration scheme are carried out in the framework of differential algebra, the phase flow $\varphi(t; x_0)$ is approximated, at each fixed time step t_i, as a Taylor expansion in x_0.

For the sake of clarity, consider the forward Euler's scheme

$$x_i = x_{i-1} + f(x_{i-1})\Delta t \tag{24}$$

and substitute the initial value with the DA identity $[x_0] = x_0 + \delta x_0$. At the first time step we have

$$[x_1] = [x_0] + f([x_0]) \cdot \Delta t. \tag{25}$$

If the function f is evaluated in the DA framework, the output of the first step, $[x_1]$, is the m-th order Taylor expansion of the flow $\varphi(t; x_0)$ in x_0 for $t = t_1$. Note that, as a result of the DA evaluation of $f([x_0])$, the $(m+1)$-tuple $[x_1]$ may include several non zeros coefficients corresponding to high order terms in δx_0. The previous procedure can be inferred through the subsequent steps. The result of the final step is the m-th order Taylor expansion of $\varphi(t; x_0)$ in x_0 at the final time t_f. Thus, the flow of a dynamical system can be approximated, at each time step t_i, as a m-th order Taylor expansion in x_0 in a fixed amount of effort.

The conversion of standard integration schemes to their DA counterparts is straightforward both for explicit and implicit solvers. This is essentially based on the substitution of the operations between real numbers with those on DA numbers. In addition, whenever the integration scheme involves iterations (e.g. iterations required in implicit and predictor-corrector methods), step size control, and order selection, a measure of the accuracy of the Taylor expansion of the flow needs to be included. For the numerical integrations presented in this paper a DA version of the Dormand and Prince (8-th order

solution for propagation, 7-th order solution for step size control) implementation of Runge-Kutta integrator is used. In this case, a weighted norm of the coefficients of the Taylor expansion is used in the step size control procedure.

Integration schemes based on DA pave the way to the nonlinear mapping of uncertainties investigated in this paper. A first example is presented hereafter about the propagation of uncertainties on initial conditions. The Taylor polynomials resulting from the use of DA-based numerical integrators expand the solution of the initial value problem presented in Eq. (23) with respect to the initial condition. Thus, the dependence of the solution $x(t)$ with respect to the initial condition is available, at a time t_i, in terms of a m-th order polynomial map $\mathcal{M}_{x_0}(\delta x_0)$, where δx_0 represents the displacement from the reference initial condition. The evaluation of the map $\mathcal{M}_{x_0}(\delta x_0)$ for a selected value of δx_0 supplies the m-th order Taylor approximation of the solution $x(t)$ at t_i corresponding to the displaced initial condition. The accuracy of the result depends on the expansion order m and the value of the displacement δx_0.

The main advantage of the DA-based approach is that there is no need to write and integrate variational equations in order to obtain high order expansions of the flow. This result is basically obtained by the substitution of operations between real numbers with those on DA numbers, and therefore the method is ODE independent. Furthermore, the efficient implementation of the differential algebra in COSY-Infinity allows us to obtain high order expansions with limited computational time.

3. Higher-order extended Kalman filters

The equations of motion describing the dynamical evolution of a spacecraft can be written in the following general form:

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}(t_{k+1}; \boldsymbol{x}_k, t_k) + \boldsymbol{w}_k, \tag{26}$$

$$\boldsymbol{z}_{k+1} = \boldsymbol{h}(\boldsymbol{x}_{k+1}, t_{k+1}) + \boldsymbol{v}_{k+1}, \tag{27}$$

where $\boldsymbol{x}_k$ is the true spacecraft state, $\boldsymbol{w}_k$ is the process noise perturbing the spacecraft state, $\boldsymbol{z}_k$ is the actual measurement, $\boldsymbol{h}$ 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 noncorrelated, 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.

Higher-order filters are nonlinear semi-analytic filtering methods which sequentially estimate the spacecraft state and the associated uncertainty. These filtering methods assume Gaussian statistics and are based on a higher-order Taylor series approach that can incorporate system nonlinearity. An overview of the HNEKF and the HAEKF filters[6] is presented hereafter.

3.1. Higher-order numerical extended Kalman filter

Consider the system model equations (26) and (27) and suppose at time t_k the state estimate has mean $\boldsymbol{m}_k^+$ and covariance matrix $\boldsymbol{P}_k^+$. Also, let $\boldsymbol{x}(t_k) = \boldsymbol{m}_k^+ + \delta\boldsymbol{x}_k$ be the true trajectory we want to estimate, with $\delta\boldsymbol{x}_k$ deviation of the estimated mean from the true trajectory. The m-th order HNEKF algorithm can be summarized as follows.

HNEKF prediction equations:

$$\begin{aligned}\left(m_{k+1}^-\right)^i =&\Phi^i\left(t_{k+1};\boldsymbol{m}_k^+,t_k\right)+\delta m_{k+1}^i\left(\delta\boldsymbol{x}_k\right)=\Phi^i\left(t_{k+1};\boldsymbol{m}_k^+,t_k\right)+\\&+\sum_{p=1}^{m}\frac{1}{p!}\Phi^{i,\gamma_1\ldots\gamma_p}_{(t_{k+1},t_k)}E\{\delta x_k^{\gamma_1}\ldots\delta x_k^{\gamma_p}\},\end{aligned}\tag{28}$$

$$\begin{aligned}\left(P_{k+1}^-\right)^{ij} =&\sum_{p=1}^{m}\sum_{q=1}^{m}\frac{1}{p!q!}\Phi^{i,\gamma_1\ldots\gamma_p}_{(t_{k+1},t_k)}\Phi^{j,\xi_1\ldots\xi_p}_{(t_{k+1},t_k)}E\{\delta x_k^{\gamma_1}\ldots\delta x_k^{\gamma_p}\delta x_k^{\xi_1}\ldots\delta x_k^{\xi_q}\}+\\&-\delta m_{k+1}^i\left(\delta\boldsymbol{x}_k\right)\delta m_{k+1}^j\left(\delta\boldsymbol{x}_k\right)+Q_k^{ij},\end{aligned}\tag{29}$$

$$\begin{aligned}\left(n_{k+1}^-\right)^i =&h^i\left(t_{k+1};\boldsymbol{m}_k^+,t_k\right)+\delta n_{k+1}^i\left(\delta\boldsymbol{x}_k\right)=h^i\left(t_{k+1};\boldsymbol{m}_k^+,t_k\right)+\\&+\sum_{p=1}^{m}\frac{1}{p!}h^{i,\gamma_1\ldots\gamma_p}_{(t_{k+1},t_k)}E\{\delta x_k^{\gamma_1}\ldots\delta x_k^{\gamma_p}\}.\end{aligned}\tag{30}$$

HNEKF update equations:

$$\begin{aligned}\left(P_{k+1}^{zz}\right)^{ij} =& \sum_{p=1}^{m}\sum_{q=1}^{m}\frac{1}{p!q!}h_{(t_{k+1},t_k)}^{i,\gamma_1\ldots\gamma_p}h_{(t_{k+1},t_k)}^{j,\xi_1\ldots\xi_p}E\{\delta x_k^{\gamma_1}\ldots\delta x_k^{\gamma_p}\delta x_k^{\xi_1}\ldots\delta x_k^{\xi_q}\}+ \\ & -\delta n_{k+1}^{i}\left(\delta\boldsymbol{x}_k\right)\delta n_{k+1}^{j}\left(\delta\boldsymbol{x}_k\right)+R_{k+1}^{ij},\end{aligned} \tag{31}$$

$$\begin{aligned}\left(P_{k+1}^{xz}\right)^{ij} =& \sum_{p=1}^{m}\sum_{q=1}^{m}\frac{1}{p!q!}\Phi_{(t_{k+1},t_k)}^{i,\gamma_1\ldots\gamma_p}h_{(t_{k+1},t_k)}^{j,\xi_1\ldots\xi_p}E\{\delta x_k^{\gamma_1}\ldots\delta x_k^{\gamma_p}\delta x_k^{\xi_1}\ldots\delta x_k^{\xi_q}\}+ \\ & -\delta m_{k+1}^{i}\left(\delta\boldsymbol{x}_k\right)\delta n_{k+1}^{j}\left(\delta\boldsymbol{x}_k\right),\end{aligned} \tag{32}$$

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^{xz}\left(\boldsymbol{P}_{k+1}^{zz}\right)^{-1}, \tag{33}$$

$$\boldsymbol{m}_{k+1}^{+} = \boldsymbol{m}_{k+1}^{-}+\boldsymbol{K}_{k+1}\left(\boldsymbol{z}_{k+1}-\boldsymbol{n}_{k+1}^{-}\right), \tag{34}$$

$$\boldsymbol{P}_{k+1}^{+} = \boldsymbol{P}_{k+1}^{-}-\boldsymbol{K}_{k+1}\boldsymbol{P}_{k+1}^{zz}\boldsymbol{K}_{k+1}^{T}. \tag{35}$$

Note that $\{\gamma_j,\xi_j\}\in\{1,...,2n\}$, where n is the dimension of the state space, $\boldsymbol{n}_{k+1}^{-}$ is the predicted measurement, $\boldsymbol{K}_{k+1}$ is the Kalman gain matrix, $\boldsymbol{P}_{k+1}^{xz}$ is the cross-covariance matrix of the state and the measurement, and $\boldsymbol{P}_{k+1}^{zz}$ is the covariance matrix of the measurements. The $\Phi_{(t_{k+1},t_k)}^{i,\gamma_1\ldots\gamma_p}$ operator includes the higher-order partials of the solution flow, which map the deviations at time m to time $(m+1)$, and $h_{(t_{k+1},t_k)}^{i,\gamma_1\ldots\gamma_p}$ includes the higher-order partials of the measurement function. These higher-order partials are called state transition tensors (STTs). In order to apply the HNEKF, the reference trajectory and its higher-order state transition tensors must be integrated for each time interval solving a system of differential equations up to the order m of the Taylor expansion. The differential equations up to the third-order deviation are given in Eq. (36)

$$\begin{cases}\dot{\Phi}^{i,a} = g^{i,\alpha}\Phi^{\alpha,a} \\ \dot{\Phi}^{i,ab} = g^{i,\alpha}\Phi^{\alpha,ab}+g^{i,\alpha\beta}\Phi^{\alpha,a}\Phi^{\beta,b} \\ \dot{\Phi}^{i,abc} = g^{i,\alpha}\Phi^{\alpha,abc}+g^{i,\alpha\beta}\left(\Phi^{\alpha,a}\Phi^{\beta,bc}+\Phi^{\alpha,ab}\Phi^{\beta,c}+\Phi^{\alpha,ac}\Phi^{\beta,b}\right)+ \\ \qquad +g^{i,\alpha\beta\gamma}\Phi^{\alpha,a}\Phi^{\beta,b}\Phi^{\gamma,c},\end{cases} \tag{36}$$

where $\boldsymbol{g}\left[t;\boldsymbol{x}\left(t\right)\right]$ represents the system dynamics vector. Once these STTs are computed, they serve a role identical to the state transition matrix except that the higher-order effects are now included, and thus the solution is nonlinear. If the common case of nonlinear measurement is considered, an equivalent system must be solved to compute the measurement function STTs.

Since this filter is based on the hypothesis of Gaussian distribution, the higher-order moments $E\{\delta x_k^{\gamma_1} \dots \delta x_k^{\gamma_p}\}$ can be completely described by the first two moments (i.e., mean and covariance) and can be calculated using the joint characteristic function (JCF). Also note that the term $\delta \boldsymbol{m}$ represents the expectation of the difference between the true and estimated state and is reset to zero at the end of each filtering step. When the Taylor expansion is arrested at the first order $m = 1$ the HNEKF becomes the EKF algorithm.

3.2. Higher-order analytic extended Kalman filter

The HAEKF can be derived from the HNEKF by assuming that the reference trajectory and the higher-order solutions (i.e., the STTs) are computed over some time span before filtering. Therefore, the HAEKF is suitable for trajectory navigation in case of missions with predetermined reference trajectories such as planetary/small-body orbit insertion or autonomous precision landing.

Suppose the STTs are computed for the time interval of $[t_0, t_f]$ and let $\bar{\boldsymbol{x}}_k = \Phi(t_k; \boldsymbol{x}_0, t_0)$ represent the reference trajectory for $t_k \in [t_0, t_f]$, where $\boldsymbol{x}_0$ has mean $\boldsymbol{m}_0^+$ and covariance $\boldsymbol{P}_0^+$. Moreover, let $\boldsymbol{x}(t_k) = \bar{\boldsymbol{x}}_k + \delta \boldsymbol{x}_k$ be the true trajectory we want to estimate.

The HAEKF filtering algorithm is similar to the HNEKF except for some aspects that can be resumed as follows. First of all, it does not require any integration in the actual filtering process because the reference trajectory and the higher-order solution are solved off-line. Moreover, the point of series expansion is now with respect to the initial reference trajectory, not the updated mean as in the HNEKF algorithm. This means that the zeroth order term that defines the predicted mean becomes $\Phi^i(t_{k+1}; \bar{\boldsymbol{x}}_k, t_k)$ and the corresponding term that define the predicted measurement becomes $h^i(\bar{\boldsymbol{x}}_{k+1}, t_{k+1})$. Also the higher-order partials are made with respect to the reference trajectory and not with respect to the current estimate. Contrary to the HNEKF, after every filtering step the term $\delta \boldsymbol{m}$ is not reset to zero, but it is set to be equal to the difference between the current estimated mean and the reference trajectory. It is clear that the convergence of the filter is guaranteed only if the true trajectory is in the convergence radius of the reference trajectory.

4. DA-based higher-order extended Kalman filters

When the nonlinearity of the considered problem is significant or when only a limited number of measurements is available, the higher-order filters presented above provide good results since they can incorporate system nonlinearity.[6] However, in many cases of practical interest, the need to derive the higher-order tensors makes them difficult to use due to computational complexity. In this section it is shown that the implementation of these filters in the DA environment can result in significant computational time savings. More in detail, two filters are discussed. First, the DA-based higher-order numerical extended Kalman filter, referred to as DAHNEKF-m throughout the paper, is introduced. Then, the DA-based higher-order analytic extended Kalman filter, referred to as DAHAEKF-m throughout the paper, is presented. In both cases m indicates the expansion order.

4.1. The DAHNEKF-m

The first step of the HNEKF algorithm requires the propagation of the state variable and its statistics from time t_k to time t_{k+1} as shown in Eq. (28) and (29). The predicted mean can be computed by solving system (36) to propagate the current estimated mean forward in time and define the STTs. The STTs are then used to calculate the deviation $\delta\boldsymbol{m}_{k+1}$ of the predicted mean from the propagated reference state. The implementation of the prediction step of the filter in the DA framework simplifies the process described above.

Let us define the current state estimate $\boldsymbol{x}_k$ as a DA variable

$$[\boldsymbol{x}_k] = \boldsymbol{m}_k^+ + \delta\boldsymbol{x}_k, \tag{37}$$

where $\boldsymbol{m}_k^+$ is the current estimated mean. If the spacecraft motion is propagated forward in time in the DA framework, the dependence of the solution on $\delta\boldsymbol{x}_k$ is carried through the algebraic operations involved in the process. The result is the Taylor expansion of the propagated state with respect to the current state and can be written as

$$[\boldsymbol{x}_{k+1}] = \bar{\boldsymbol{x}}_{k+1} + \mathcal{M}_{\boldsymbol{x}_{k+1}}(\delta\boldsymbol{x}_k) = \sum_{p_1+\cdots+p_n\leq m} \boldsymbol{c}_{p_1\ldots p_n} \cdot \delta x_{k,1}^{p_1} \cdots \delta x_{k,n}^{p_n}, \tag{38}$$

where $\bar{\boldsymbol{x}}_{k+1}$ is the zeroth order term of the expansion map, and $\boldsymbol{c}_{p_1\ldots p_n}$ are the Taylor coefficients of the resulting Taylor polynomial

$$\boldsymbol{c}_{p_1\ldots p_n} = \frac{1}{p_1!\cdots p_n!} \cdot \frac{\partial^{p_1+\cdots+p_n}\boldsymbol{\Phi}}{\partial x_1^{p_1}\cdots\partial x_n^{p_n}}. \tag{39}$$

The coefficients $\boldsymbol{c}_{p_1 \ldots p_n}$ are the terms that relate deviations in the initial conditions to the state at some future time and therefore exactly correspond to the STTs defined in the original formulation of the HNEKF filter. The evaluation of Eq. (38) for a selected value of $\delta \boldsymbol{x}_k$ supplies the m-th order Taylor approximation of the state $\boldsymbol{x}_{k+1}$ corresponding to the displaced current state. Of course, the accuracy of the expansion map is function of the expansion order and can be controlled by tuning it. The Taylor series in the form (38) can be used to efficiently compute the propagated statistics.

Let us recall that for a generic scalar random variable z with probability distribution function (pdf) $g(z)$ the first two moments can be written as

$$\begin{cases} m = E\{z\} \\ P = E\{(z-m)^2\}, \end{cases} \tag{40}$$

where m is the mean value, and P is the covariance,[15] and the expectation value of z is defined as

$$E\{z\} = \int_{-\infty}^{+\infty} z g(z) dz. \tag{41}$$

So, the moments of the propagated pdf for problem (26) can be computed by applying the multivariate form of Eq. (40) to the Taylor expansion (38). The result becomes

$$(m_{k+1}^-)^i = E\{[x_{k+1}^i]\} = \sum_{p_1+\cdots+p_n \leq m} \boldsymbol{c}_{i,p_1 \ldots p_n} E\{\delta x_{k,1}^{p_1} \cdots \delta x_{k,n}^{p_n}\}, \tag{42}$$

$$\begin{aligned} (P_{k+1}^-)^{ij} =& E\{([x_{k+1}^i] - (m_{k+1}^-)^i)([x_{k+1}^j] - (m_{k+1}^-)^j)\} = \\ & \sum_{\substack{p_1+\cdots+p_n \leq m, \\ q_1+\cdots+q_n \leq m}} \boldsymbol{c}_{i,p_1 \ldots p_n} \boldsymbol{c}_{j,q_1 \ldots q_n} E\{\delta x_{k,1}^{p_1+q_1} \cdots \delta x_{k,n}^{p_n+q_n}\}, \end{aligned} \tag{43}$$

where $\boldsymbol{c}_{i,p_1 \ldots p_n}$ are the Taylor coefficients of the Taylor polynomial describing the i-th component of $[\boldsymbol{x}_{k+1}]$. Note that in the covariance matrix formula the coefficients $\boldsymbol{c}_{i,p_1 \ldots p_n}$ and $\boldsymbol{c}_{j,q_1 \ldots q_n}$ already include the subtraction of the mean terms. The coefficients of the second order moment are computed by implementing the required operations (i.e., $([x_{k+1}^i] - (m_{k+1}^-)^i)([x_{k+1}^j] - (m_{k+1}^-)^j)$) on Taylor polynomials in the DA framework. The expectation values on the right side of Eq.s (42) and (43) are function of $p(\boldsymbol{x}_k)$. It follows that if the probability distribution at time m is known, the predicted

mean and covariance can be calculated. The number of monomials for which it is necessary to compute the expectation increases with the order of the Taylor expansion and, of course, with the order of the moment we want to compute. Note that, at this time, no hypothesis on the initial pdf has been made. Thus, the method can be applied independently of the considered variable distribution. However, the Gaussian approximation is widely used in navigation and real-time nonlinear filtering to represent the current state variable distribution during the filtering process. In this case, as already anticipated, the higher-order moments can be completely described by the first two moments and can be calculated using the JCF.

Also the predicted measurement (30) and the update step of the filtering algorithm can benefit from their implementation in the DA framework. In fact evaluating the measurement equation (27) in the DA framework allow us to find the Taylor approximation of the measurement at time t_{k+1} corresponding to the displaced state $\delta\boldsymbol{x}_{k+1}$

$$[\boldsymbol{z}_{k+1}] = \boldsymbol{h}\left(\left[\boldsymbol{x}_{k+1}^{-}\right], t_{k+1}\right) = \bar{\boldsymbol{z}}_{k+1} + \mathcal{M}_{\boldsymbol{z}_{k+1}}\left(\delta\boldsymbol{x}_k\right). \tag{44}$$

The predicted measurement $\left(n_{k+1}^{-}\right)^i$ can be computed by calculating the expectation value of (44), while the cross-covariance matrix of the state and the measurement and the covariance matrix of the measurements can be computed as shown in Eq.s (45) and (46), respectively.

$$\left(P_{k+1}^{xz}\right)^{ij} = E\{([x_{k+1}^i] - (m_{k+1}^{-})^i)([z_{k+1}^j] - (n_{k+1}^{-})^j)\} \tag{45}$$

$$\left(P_{k+1}^{zz}\right)^{ij} = E\{([z_{k+1}^i] - (n_{k+1}^{-})^i)([z_{k+1}^j] - (n_{k+1}^{-})^j)\} \tag{46}$$

So, the resulting DAHNEKF-m algorithm can be summarized as presented in Eq. (47).

$$\begin{aligned}
(m_{k+1}^{-})^i &= E\{[x_{k+1}^i]\},\\
(P_{k+1}^{-})^{ij} &= E\{([x_{k+1}^i] - (m_{k+1}^{-})^i)([x_{k+1}^j] - (m_{k+1}^{-})^j)\},\\
(n_{k+1}^{-})^i &= E\{[z_{k+1}^i]\},\\
\left(P_{k+1}^{xz}\right)^{ij} &= E\{([x_{k+1}^i] - (m_{k+1}^{-})^i)([z_{k+1}^j] - (n_{k+1}^{-})^j)\},\\
\left(P_{k+1}^{zz}\right)^{ij} &= E\{([z_{k+1}^i] - (n_{k+1}^{-})^i)([z_{k+1}^j] - (n_{k+1}^{-})^j)\},\\
\boldsymbol{K}_{k+1} &= \boldsymbol{P}_{k+1}^{xz}\left(\boldsymbol{P}_{k+1}^{zz}\right)^{-1},\\
\boldsymbol{m}_{k+1}^{+} &= \boldsymbol{m}_{k+1}^{-} + \boldsymbol{K}_{k+1}\left(\boldsymbol{z}_{k+1} - \boldsymbol{n}_{k+1}^{-}\right),\\
\boldsymbol{P}_{k+1}^{+} &= \boldsymbol{P}_{k+1}^{-} - \boldsymbol{K}_{k+1}\boldsymbol{P}_{k+1}^{zz}\boldsymbol{K}_{k+1}^{T}.
\end{aligned} \tag{47}$$

The DAHNEKF-m removes the need to solve for two systems (one for the state STT and one for the measurement STT) of differential equations up to the m-th order deviation, i.e., the order of the Taylor series expansion. The integration of these complex systems is replaced by just one integration (Eq. (38)) and one function evaluation (Eq. (44)) in differential algebra. Thanks to the efficiency of DA implementation, the DAHNEKF-m may be suitable for on-line filtering applications such as autonomous trajectory navigation. This approach not only consistently speeds up the filtering process, but it is also very flexible about changes in the system dynamics or initial conditions. If, for instance, some changes in the problem dynamics occur, the HNEKF requires to rewrite the variational equations, while in the DAHNEKF-m we simply need to update the system model (26).

4.2. The DAHAEKF-m

Differential algebra provides an elegant solution in case of missions with predetermined reference trajectories. In these cases, in fact, the DAHNAEKF-m requires only polynomial evaluations, as there is no need for integration in the actual filtering process. The reference trajectory is computed in the DA framework over some time span before filtering, obtaining the Taylor expansion of the reference solution with respect to the initial reference deviated state.

Suppose the reference trajectory is integrated for the time interval $[t_0, t_f]$ and let $\left[\boldsymbol{x}_{k+1}^{ref}\right] = \mathcal{M}_{(t_{k+1},t_k)}(\delta\boldsymbol{x}_k^{ref})$ represent the reference trajectory expansion map for time t_{k+1} (as shown in Figure 2), where $\boldsymbol{x}_k^{ref}$ has mean $\boldsymbol{m}_k^+$ and covariance $\boldsymbol{P}_k^+$. Since this integration is performed offline, the Taylor expansion order can be kept high. Once the expansion map related to the reference solution is known, the actual filtering process can start. The DAHAEKF-m algorithm can be summarized as follows:

1. Compute the first two moments of the propagated pdf (i.e., the predicted mean and covariance) calculating the expectation value of a proper form of the l-th ($l = 1,2$) power of expansion map $\mathcal{M}_{(t_{k+1},t_k)}(\delta\boldsymbol{x}_k^{ref})$ truncated at order m;
2. Compute the Taylor expansion of the measurement with respect to the deviated initial reference state $[\boldsymbol{z}_{k+1}] = \mathcal{M}_{\boldsymbol{z}_{k+1}}(\delta\boldsymbol{x}_k^{ref})$;
3. Compute the predicted measurement calculating the expectation value of expansion map $\mathcal{M}_{\boldsymbol{z}_{k+1}}(\delta\boldsymbol{x}_k^{ref})$;

4. Compute $\boldsymbol{P}_{zz}$ calculating the expectation value of the second power of expansion map $\mathcal{M}_{\boldsymbol{z}_{k+1}}(\delta \boldsymbol{x}_k^{ref})$;
5. Compute $\boldsymbol{P}_{xz}$ calculating the expectation value of the product between expansion maps $\mathcal{M}_{(t_{k+1},t_k)}(\delta \boldsymbol{x}_k^{ref})$ and $\mathcal{M}_{\boldsymbol{z}_{k+1}}(\delta \boldsymbol{x}_k^{ref})$.
6. Procede with the update filtering step as in the HAEKF.

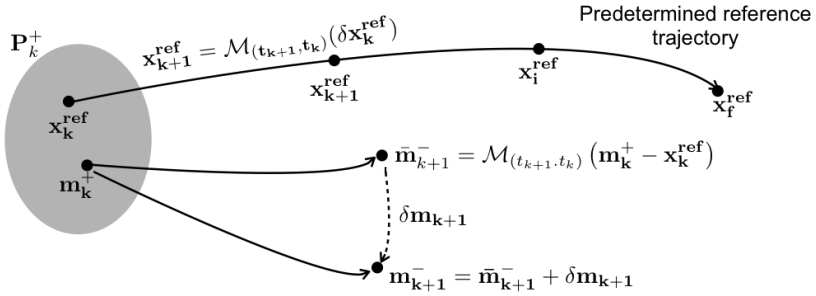


Figure 2: Scheme of the DAHNAEKF-m prediction step. The exponential map related to the predetermined trajectory is computed offline while only map evaluations are performed online.

5. Numerical examples

The DAHNEKF-m and DAHAEKF-m filters are tested and applied to some typical problems in celestial mechanics. In this section, two main cases are presented. First of all the case of a sun-Earth halo orbit about the L_1 point is considered. This problem is the same used to study the performance of the HNEKF and HAEKF[6] and it is here considered to demonstrate the equivalence of the two classes of filters (i.e., standard higher-order and DA-based higher-order methods) in terms of accuracy. This example is also used to show the computational efficiency of the DA-based methods. On the other hand, the second numerical example is the case of a Earth orbiting satellite with realistic orbit uncertainties and nonlinear measurements.

5.1. Sun-Earth Halo orbit about the L_1 point

The governing equations of motion based on the circular restricted three body problem (CR3BP), in nondimensional form, are given as

$$\ddot{x} - 2\dot{y} = \frac{\partial U}{\partial x}, \tag{48}$$

$$\ddot{y} + 2\dot{x} = \frac{\partial U}{\partial y}, \tag{49}$$

$$\ddot{z} = \frac{\partial U}{\partial z}, \tag{50}$$

where

$$U = \frac{(1-\mu)}{r_1} + \frac{\mu}{r_2} + \frac{(x^2+y^2)}{2}, \tag{51}$$

$$r_1 = [(x+\mu)^2 + y^2 + z^2]^{1/2}, \tag{52}$$

$$r_2 = [(x-1+\mu)^2 + y^2 + z^2]^{1/2}. \tag{53}$$

Here, U is the CR3BP potential and $\mu = \mu_e/(\mu_s + \mu_e)$, where μ_e and μ_s are the gravitational parameter of the Earth and the Sun respectively. The initial position and velocity assumed for the analysis are

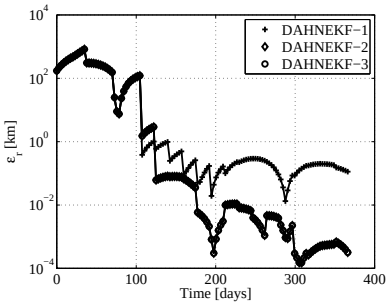
$$\begin{aligned} \boldsymbol{r}_0 &= [0.988884102845168, 0.0, 0.000921858528329094]^T \\ \boldsymbol{v}_0 &= [0.0, 0.00893471471659142, 0.0]^T, \end{aligned} \tag{54}$$

where the length units are scaled by $l = 1$ AU $= 1.49597870691 \times 10^8$ km, and the time by the constant $\tau = 1/1.991 \times 10^{-7}$ s. The considered measurement model is

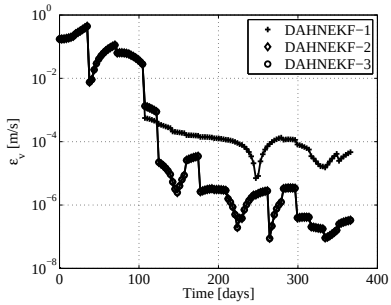
$$z_k = y_k + v_k, \tag{55}$$

where y_k is the second position component of the state vector and v_k represents the measurement error, assumed to be 0.1 m for each range measurement. The considered initial guess is off from the true trajectory by 100 km for the position components and 0.1 m/s for the velocity components so that they lie on the boundary of the initial 1-σ ellipsoid. Measurement is performed with a 20-day increment. No process noise is considered.

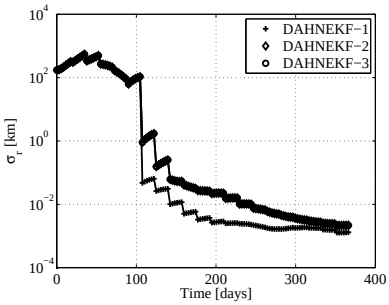
Simulation results are illustrated in Figures 3 and 4. First of all, it is important to observe that the obtained accuracy is consistent with the results obtained from the implementation of the HNEKF and HAEKF by Park and Scheeres.[6] This is an expected result since the DA-based methods are an efficient way from a computational point of view to perform higher-order filtering, but they do not change the approach to the problem and the filtering process methodology of the HNEKF and HAEKF. The results show that the higher-order filters, $m \in \{2, 3\}$, provide superior filter performance over the first-order case, which also overestimates the uncertainties. Moreover, it is observed that the second-order effect contains most of the system nonlinearity



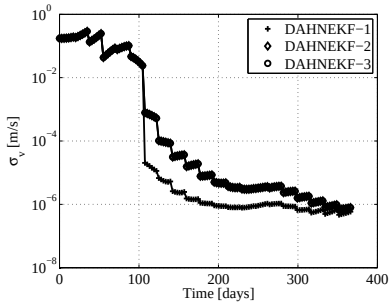
(a) Position absolute errors, i.e., $\epsilon_r = \sqrt{x^2 + y^2 + z^2}$



(b) Velocity absolute errors, i.e., $\epsilon_v = \sqrt{v_x^2 + v_y^2 + v_z^2}$

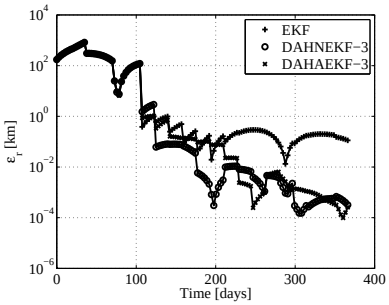


(c) Position root-sum-square errors, i.e., $\sigma_r = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2}$

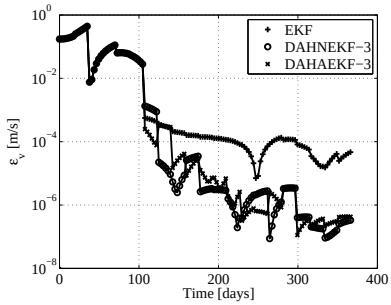


(d) Velocity root-sum-square errors, i.e., $\sigma_v = \sqrt{\sigma_{vx}^2 + \sigma_{vy}^2 + \sigma_{vz}^2}$

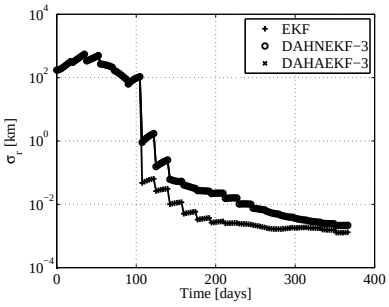
Figure 3: Results obtained using the DAHNEKF-m, with $m = 1, 2, 3$, for the sun-Earth halo orbit about L_1 point test case.



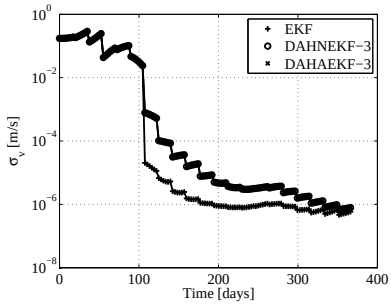
(a) Position absolute errors, i.e., $\epsilon_r = \sqrt{x^2 + y^2 + z^2}$



(b) Velocity absolute errors, i.e., $\epsilon_v = \sqrt{v_x^2 + v_y^2 + v_z^2}$



(c) Position root-sum-square errors, i.e., $\sigma_r = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2}$



(d) Velocity root-sum-square errors, i.e., $\sigma_v = \sqrt{\sigma_{vx}^2 + \sigma_{vy}^2 + \sigma_{vz}^2}$

Figure 4: Results obtained comparing the EKF, the DAHNEKF-3 and the DAHAEKF-3, for the sun-Earth halo orbit about L_1 point test case.

indicating that in this example the second-order filter is sufficient for the implementation of an accurate nonlinear filter.

The simulation results show that the DA-based higher-order filters are characterized by the same accuracy of their counterparts in real algebra. An important consideration on the computational effort can be made at this point. Let n be the dimension of the state vector to be estimated. In the frame of variational approach the number of equations that must be integrated together are $\sum_{q=1}^{m+1} n^q$.[15] If we assume that the integration time scales linearly with the number of equations, the ratio R_τ of the computational time between a m-th order computation and its zeroth order counterpart is $R_\tau = 1 + \sum_{q=1}^{m} n^q$. As an example, a second order computation with $n = 6$ implies the integration of 258 equations and $R_\tau = 43$. At third order the number of equations rises to 1554 and $R_\tau = 259$. In Figure 5 the values of R_τ associated to DA expansion of the flow (represented in Eq. (38)) of the ODE system defined by equations (48)–(49) are compared with those of a variational approach. It is apparent that the DA method is orders of magnitudes more efficient than the variational one. Furthermore, the DA method is application independent and the intricate task of finding the analytical expression of the variational equations is avoided.

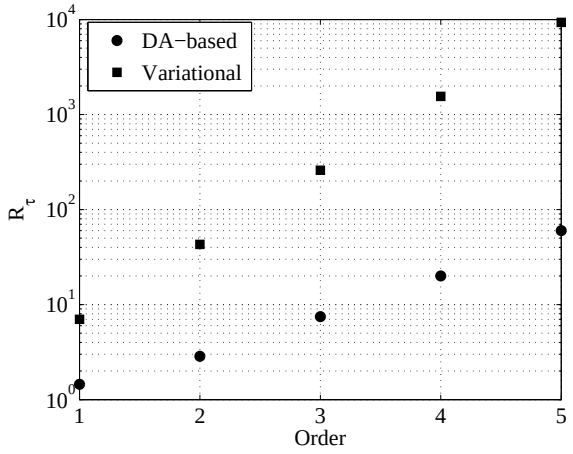


Figure 5: Ratio of the computational time between a m-th order expansion of the flow and its zeroth order counterpart.

In addition, to authors knowledge, the efficiency of DA techniques is far better than any other automatic differentiation method suitable for the computation of the m-th order Taylor approximation of the flow of ordinary

differential equations (see for example[8]). This is also confirmed by the statement *Without introducing new approaches to handle the multidimensional algebra and calculus,...* the problem *...becomes computationally cumbersome above second-order* that appears in the conclusive section of.[7]

5.2. Two-body problem: Earth's artificial satellite

The orbit of an Earth's artificial satellite is considered to be estimated. The second-order differential equation governing the motion is

$$\frac{d\dot{\boldsymbol{r}}}{dt} = -\frac{\mu}{r^3}\boldsymbol{r}, \tag{56}$$

where $\boldsymbol{r}$ is the position vector of the spacecraft and μ is the Earth gravitational parameter. It is assumed that there is no external disturbing force for the system except the gravitational force between the Earth and the spacecraft. The initial true position and velocity assumed for this analysis are

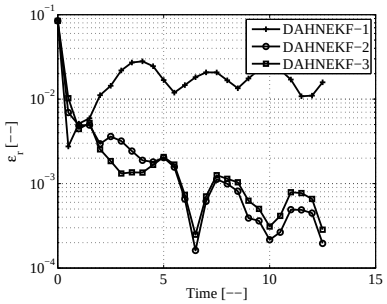
$$\boldsymbol{x}_0 = \begin{pmatrix} \boldsymbol{r}_0 \\ \boldsymbol{v}_0 \end{pmatrix} = \begin{pmatrix} -0.68787 \\ -0.39713 \\ +0.28448 \\ -0.51331 \\ +0.98266 \\ +0.37611 \end{pmatrix}, \tag{57}$$

where the length units are scaled by the orbit semi- major axis (a=8788 km) and the time by $\sqrt{\frac{a^3}{\mu}}$. The initial estimates for the state are 10% off from the true initial state values shown in Eq. 57. The adopted initial error covariance is a diagonal matrix with variance 0.01 for the position vector components and 10^{-4} for the velocity vector components. The measurements used in the simulation are the radial position of the spacecraft with respect to the Earth and the line of sight directions to the planet, as shown in Eq. 58

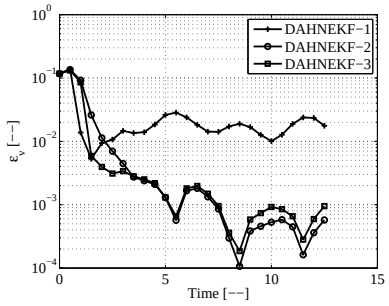
$$\begin{aligned} z_1 &= r + v_1 \\ z_2 &= \arctan\left(\frac{y}{x}\right) + v_2 \\ z_3 &= \arcsin\left(\frac{z}{r}\right) + v_3, \end{aligned} \tag{58}$$

where v_i, $i = 1, 2, 3$, represents the measurement noise. The standard deviation of the measurement error for the radial position is assumed to be

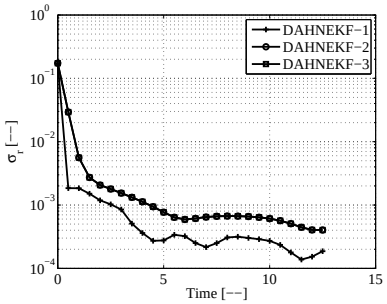
10^{-3} km. The angle error is assumed to be 1.745 10^{-6} rad following the conventional measurement accuracy for the directional measurement. In one orbit a total number of 12 measurements separated by equal time intervals is considered. No process noise is included.



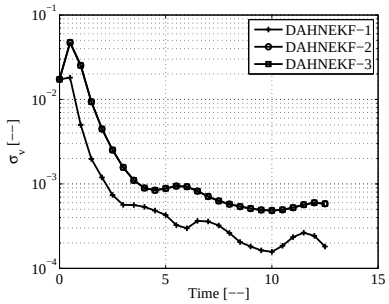
(a) Position absolute errors, i.e., $\epsilon_r = \sqrt{x^2 + y^2 + z^2}$



(b) Velocity absolute errors, i.e., $\epsilon_v = \sqrt{v_x^2 + v_y^2 + v_z^2}$



(c) Position root-sum-square errors, i.e., $\sigma_r = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2}$



(d) Velocity root-sum-square errors, i.e., $\sigma_v = \sqrt{\sigma_{vx}^2 + \sigma_{vy}^2 + \sigma_{vz}^2}$

Figure 6: Results obtained using the DAHNEKF-m, with $m = 1, 2, 3$, for the Earth's artificial satellite test case.

Looking at the simulation results presented in Figure 6, we see that as the filter order increases, the estimation accuracy tends to improve. More in detail, higher-order filters, $m \in \{2, 3\}$, provide superior filter performance over the first-order case, which overestimates the uncertainties. This not only explains the importance of nonlinear trajectory navigation, but also shows how the higher-order filters can extract more information from the available nonlinear measurements compared to the first-order filter. The results of the DAHAEKF-m are identical in terms of accuracy to the ones obtained from

the DAHNEKF-m.

In case of nonlinear measurements the DA-based approach significantly simplifies the problem compared to the standard higher-order methods. Indeed, in such a case, the computation of the higher-order partials of the measurement function as done in the HNEKF is quite complex and very time-consuming. On the other hand, using differential algebra we automatically obtain the higher-order partials along with the measurement function evaluation. The DAHNEKF-m and DAHAEKF-m become more and more efficient compared to the HNEKF and HAEKF as the complexity and nonlinearity of the dynamical system and measurement equation increase.

6. Conclusions

In this paper the problem of nonlinear filtering for spacecraft navigation has been addressed. We derived two filters, called the differential algebra-based higher-order extended Kalman filter (DAHNEKF-m) and the differential algebra-based higher-order analytic Kalman filter (DAHAEKF-m), by implementing the higher-order extended Kalman filter and the higher-order analytic Kalman filter in a differential algebra framework. It has been demonstrated that working in the differential algebra framework significantly reduces the complexity and the computational burden related to the standard higher-order approaches. The algorithms have been tested on some typical problems in spacecraft navigation such as the two-body problem and the circular restricted three-body problem to demonstrate their effectiveness and good performance. The simulation results show that higher orders can improve the accuracy of the state determination since they can extract, from the available nonlinear measurements, more accurate information about the state of the vehicle than low order filters. Numerical simulations show the good performance of the filters in case of both complex dynamics and highly nonlinear measurement problems.

References

[1] Bierman, G.J., Factorization methods for discrete sequential estimation, Vol. 128, Academic Press, 1977.

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

[3] Julier, S.J. and Uhlmann, J.K., Unscented filtering and nonlinear estimation, Proceedings of the IEEE, Vol. 92, No. 3, pp. 401–422, 2004.

[4] Julier, S.J. and Uhlmann, J.K. and Durrant-Whyte, H.F., A new approach for filtering nonlinear systems, American Control Conference, Vol. 3, pp. 1628–1632, Proceedings of the IEEE, 1995.

[5] Park, R.S. and Scheeres, D.J., Nonlinear mapping of Gaussian statistics: theory and applications to spacecraft trajectory design, Journal of Guidance Control and Dynamics, Vol. 29, No. 6, pp. 1367–1375, American Institute of Aeronautics and Astronautics, 2006

[6] Park, R.S. and Scheeres, D.J., Nonlinear semi-analytic methods for trajectory estimation, Journal of guidance, control, and dynamics, Vol. 30, No. 6, pp. 1668–1676, American Institute of Aeronautics and Astronautics, 2007.

[7] Majji, M., Junkins, J.L., and Turner J.D., A high order method for estimation of dynamic systems, The Journal of the Astronautical Sciences, Vol. 56, No. 3, pp. 1–32, 2008.

[8] Vittaldev, V., Russell, R.P., Arora, N., and Gaylor, D., Second-order Kalman filters using multi-complex step derivatives, Paper AAS 12-204, AAS/AIAA Space Flight Mechanics Meeting, Charleston, SC, Jan 2012.

[9] Berz, M., Differential Algebraic Techniques, Entry in Handbook of Accelerator Physics and Engineering, World Scientific, New York, 1999.

[10] Berz, M. and Makino, K. and Shamseddine, K., Modern map methods in particle beam physics, Vol. 108, Academic Press, 1999.

[11] M. Berz and K. Makino, COSY INFINITY version 9 reference manual, MSU Report MSUHEP060803, East Lansing, MI 48824, 2006.

[12] Armellin, R. and Di Lizia, P. and Bernelli-Zazzera, F. and Berz, M., Asteroid close encounters characterization using differential algebra: the case of Apophis, Celestial Mechanics and Dynamical Astronomy, Vol. 107, No. 4, pp. 451–470, Springer, 2010.

[13] Berz, M., The new method of TPSA algebra for the description of beam dynamics to high orders, Technical Report AT-6: ATN-86-16, Los Alamos National Laboratory, 1986.

[14] 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, pp. 431–436, Elsevier, 1987.

[15] Casella, G., and Berger, R.L., Statistical inference, pp. 55–68, Duxbury Press, 2001.