Noname manuscript No.
(will be inserted by the editor)

Maximum A Posteriori Estimation of Hamiltonian Systems with High Order Taylor Polynomials

Simone Servadio · Renato Zanetti · Roberto Armellin

Received: date / Accepted: date

Abstract This paper presents a new approach to Maximum A Posteriori (MAP) estimation for Hamiltonian dynamic systems. By representing probability density functions through Taylor polynomials and using Differential Algebra techniques, this work proposes to derive the MAP estimate directly from high order polynomials. The polynomial representation of the posterior probability density function leads to an accurate approximation of the true *a posterior* distribution, that describes the uncertainties of the state of the system. The new method is applied to a demonstrative orbit determination problem.

Keywords Filtering Problem · Halo Orbits · Maximum A Posteriori · Nonlinear Estimation · Orbit Determination

1 Introduction

The Maximum A Posteriori (MAP) estimate is given by the peak of the probability density function conditioned on the value of the measurements, also

S. Servadio
Ph.D. Candidate
Aerospace Engineering and Engineering Mechanics
The University of Texas at Austin, Austin, Texas
E-mail: simo_serva@utexas.edu

R. Zanetti
Assistant Professor
Aerospace Engineering and Engineering Mechanics
The University of Texas at Austin, Austin, Texas
E-mail: renato@utexas.edu

R. Armellin
Professor of Te Pūnaha Ātea - Auckland Space Institute
The University of Auckland, Auckland, New Zealand
E-mail: roberto.armellin@auckland.ac.nz

known as the posterior distribution, which is typically calculated using Baye's rule. In the well-known linear and Gaussian case, the posterior distribution remains Gaussian at all times and it can be fully described by its mean and covariance matrix, which are calculated using the Kalman filter equations [1, 2]. For this type of problems, Linear Minimum Mean Square Error (LMMSE) estimation coincides with the MAP estimation. Most problems of practical interest to aerospace engineering applications, such as orbit determination [3], contain nonlinearities in the dynamics and/or measurements which inevitably result in a non-Gaussian posterior distribution. In these situations, LMMSE and MAP produce different estimates and a closed form solution of either is typically unavailable.

Many techniques have been developed to deal with nonlinearities in the estimation problem. The most exploited estimation algorithm is the Extended Kalman Filter (EKF) [4]. The EKF is based on linearization of the dynamics and measurement equations around the most current estimate and applies the Kalman filter equations to the linearized system. In problems with high nonlinearities, including certain orbit determination applications, the linearization assumption may fail to provide a valid estimate [5]. Other approaches for estimation of nonlinear systems have been implemented, such as the Unscented Kalman filter (UKF) [6,7]. Through the true nonlinear propagation of carefully chosen deterministic points (sigma points), the UKF can achieve superior performance with respect to the EKF.

Park and Scheeres [8,9] developed higher order filters that uses state transition tensors (STT) in the filter's prediction step that fully relies on the non linear mapping of the means and covariance matrices to more accurately represent the uncertainty with respect to the EKF. Valli et al.[10] recreated Park and Scheeres's work using Differential Algebra (DA) [11,12], thus eliminating the need to evaluate high order tensors directly.

This work introduces a new nonlinear filter for Hamiltonian systems [13] that approximates the probability distribution functions through Taylor expansion polynomials using DA. The time propagation of the Hamiltonian dynamics and the measurement update phases of the filter operate directly on the probability density function (PDF). The MAP estimate is then found by polynomial maximization. The Taylor polynomials are kept centered on the most recent estimate. Lastly, an estimate of the estimate's uncertainty is provided through Monte Carlo integration techniques.

2 Maximum A Posteriori Estimation

Let $p_{\mathbf{x}}(\mathbf{x})$ be the probability density function (PDF) of a random vector $\mathbf{x}$, called the state, that it is not directly sample-able and whose outcome we are interested in estimating; and let $\mathbf{y}$ be another random vector, called the measurement, that possesses a joint distribution with $\mathbf{x}$ and that it is available to be sampled. The conditional distribution of $\mathbf{x}$ given $\mathbf{y}$, denoted as $p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y})$

is given by Baye's rule

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) = \frac{p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})\ p_{\mathbf{x}}(\mathbf{x})}{p_{\mathbf{y}}(\mathbf{y})} \tag{1}$$

where $p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y})$ is called posterior distribution, $p_{\mathbf{x}}(\mathbf{x})$ prior distribution, and $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) = L(\mathbf{x}|\mathbf{y})$ likelihood function.

The Maximum A Posteriori estimate $\hat{\mathbf{x}}$ is the, possibly not unique, value of $\mathbf{x}$ that maximizes the posterior distribution

$$\hat{\mathbf{x}} = \max_{\mathbf{x}} p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) \tag{2}$$

We notice that the denominator of Eq. (1) does not contain $\mathbf{x}$, therefore an equivalent optimization problem is to maximize the un-normalized posterior distribution, i.e. the joint distribution of $\mathbf{x}$ and $\mathbf{y}$: $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y}) = p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})\ p_{\mathbf{x}}(\mathbf{x})$.

$$\hat{\mathbf{x}} = \max_{\mathbf{x}} p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y}) \tag{3}$$

Rather than maximizing the joint distribution directly, it is often convenient to maximize its logarithm:

$$\hat{\mathbf{x}} = \max_{\mathbf{x}} \Big(\log p_{\mathbf{x}}(\mathbf{x}) + \log p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) \Big) \tag{4}$$

where $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})$ is the likelihood $\log L(\mathbf{y}|\mathbf{x})$.

3 Uncertainty Evolution of Hamiltonian Systems

Assume the state vector time evolution is governed by the following ordinary differential equation (ODE)

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t,\mathbf{x}); \qquad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{5}$$

let's denote the solution flow as $\mathbf{x}(t) = \boldsymbol{\phi}(\mathbf{x}_0; t, t_0)$; clearly

$$\frac{\partial}{\partial t}\boldsymbol{\phi}(\mathbf{x}_0; t, t_0) = \mathbf{f}(t, \boldsymbol{\phi}(\mathbf{x}_0; t, t_0)) \tag{6}$$

The solution flow has the following properties

$$\boldsymbol{\phi}(\mathbf{x}_0; t_0, t_0) = \mathbf{x}_0 \tag{7}$$

$$\boldsymbol{\phi}(\boldsymbol{\phi}(\mathbf{x}_0; t, t_0); t_0, t) = \mathbf{x}_0 \tag{8}$$

this second property just states that the functional inverse is obtained by inverting the time arguments (exactly like the state transition matrix)

$$\mathbf{x}_f = \boldsymbol{\phi}(\mathbf{x}_0; t_f, t_0) \tag{9}$$

$$\mathbf{x}_0 = \boldsymbol{\phi}(\mathbf{x}_f; t_0, t_f) \tag{10}$$

From

$$\boldsymbol{\phi}(\boldsymbol{\phi}(\mathbf{x}_0;t,t_0);t_0,t) = \mathbf{x}_0 \tag{11}$$

we have that

$$\begin{aligned}\frac{d}{dt}\boldsymbol{\phi}(\boldsymbol{\phi}(\mathbf{x}_0;t,t_0);t_0,t) &= \mathbf{0} \\ = \frac{\partial}{\partial t}\boldsymbol{\phi}(\boldsymbol{\phi}(\mathbf{x}_0;t,t_0);t_0,t) &+ \frac{\partial}{\partial \xi}\boldsymbol{\phi}(\boldsymbol{\xi};t_0,t)\frac{\partial}{\partial t}\boldsymbol{\phi}(\mathbf{x}_0;t,t_0)\end{aligned} \tag{12}$$

or

$$\frac{\partial}{\partial t}\boldsymbol{\phi}(\mathbf{x};t_0,t) = -\left.\frac{\partial}{\partial \xi}\boldsymbol{\phi}(\boldsymbol{\xi};t_0,t)\right|_{\boldsymbol{\xi}=\mathbf{x}} \mathbf{f}(t,\mathbf{x}) \tag{13}$$

Taking the initial condition $\mathbf{x}_0$ as a random vector with PDF $p_{\mathbf{x}_0}(\mathbf{x}_0)$, the PDF evolves in time according to the Fokker-Plank-Kolmogorov (FPK) [14] equation without diffusion (also known as Louville equation)

$$\begin{aligned}\frac{\partial p_{\mathbf{x}(t)}(\mathbf{x},t)}{\partial t} &= -\sum_{i=1}^{n}\frac{\partial\Big(f_i(\mathbf{x},t)\ p_{\mathbf{x}(t)}(\mathbf{x},t)\Big)}{\partial x_i} \\ = -\sum_{i=1}^{n} f_i(\mathbf{x},t)\ \frac{\partial p_{\mathbf{x}(t)}(\mathbf{x},t)}{\partial x_i} &- \sum_{i=1}^{n} p_{\mathbf{x}(t)}(\mathbf{x},t)\ \frac{\partial f_i(\mathbf{x},t)}{\partial x_i} \\ = -\frac{\partial p_{\mathbf{x}(t)}(\mathbf{x},t)}{\partial \mathbf{x}}\ \mathbf{f}(\mathbf{x},t) &- p_{\mathbf{x}(t)}(\mathbf{x},t)\ \ \mathrm{trace}\left(\frac{\partial \mathbf{f}(\mathbf{x},t)}{\partial \mathbf{x}}\right)\end{aligned} \tag{14}$$

For Hamiltonian systems the Jacobian of $\mathbf{f}$ is traceless and we are finally left with the well known result that the total time derivative of the PDF is zero for a Hamiltonian system:

$$\frac{\partial p_{\mathbf{x}(t)}(\mathbf{x},t)}{\partial t} + \frac{\partial p_{\mathbf{x}(t)}(\mathbf{x},t)}{\partial \mathbf{x}}\ \dot{\mathbf{x}} = \frac{dp_{\mathbf{x}(t)}(\mathbf{x},t)}{dt} = 0 \tag{15}$$

hence for all t

$$p_{\mathbf{x}(t)}(\mathbf{x},t) = p_{\mathbf{x}(t_0)}(\boldsymbol{\phi}(\mathbf{x};t_0,t),t_0) = p_{\mathbf{x}_0}(\boldsymbol{\phi}^{-1}(\mathbf{x};t,t_0)) \tag{16}$$

This is easy to show since

$$\begin{aligned}\frac{dp_{\mathbf{x}(t)}(\mathbf{x},t)}{dt} &= \frac{dp_{\mathbf{x}_0}(\boldsymbol{\phi}(\mathbf{x};t_0,t))}{dt} \\ &= \frac{\partial p_{\mathbf{x}_0}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\left(\frac{\partial \boldsymbol{\phi}(\mathbf{x};t_0,t)}{\partial t} + \frac{\partial \boldsymbol{\phi}(\mathbf{x};t_0,t)}{\partial \mathbf{x}}\mathbf{f}(t,\mathbf{x})\right)\end{aligned} \tag{17}$$

However, the term in parenthesis is identically zero because of Eq. (12), so the FPK equation is satisfied.

4 Differential Algebra

The key idea of DA is to define algebraic operations between polynomials similar to those commonly used with real numbers [15] by suppling the tools to compute the derivatives of functions within a computer environment [16]. More specifically, by substituting the classical implementation of real algebra with the implementation of a new algebra of Taylor polynomials, any function of n variables is expanded into its Taylor polynomial up to an arbitrary, user-defined, order c. A possible representation of analytic functions in a computer environment is an approximation achieved with a collection of points, Differential Algebra is a different representation of functions given by the coefficients of monomial basis functions. Using this framework in a computer environment allows to treat functions and their operations similarly to how real numbers are typically handled. Real numbers cannot be treated, in general, in a computer environment, and they are approximated via truncation by floating point (FP) numbers with a finite number of digits. Referring to Figure 1 [17,18], let us consider two real numbers a and b and their floating point approximation $\bar{a}$ and $\bar{b}$ respectively. Given any operation "$*$" in the set of real numbers, an adjoint operation "$\circledast$" is defined in the set of FP numbers such that the diagram in the figure commutes. 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 result into its FP representation. In a similar way, consider the right part of the figure, where two c times differentiable functions f and g are given. In the DA framework, the computer operates on them using their cth order Taylor expansions, F and G, respectively. The operations performed in the space of Taylor polynomials return the same result as operating on f and g in the original space and then extracting the resulting Taylor series expansion. Therefore, once a center point for the Taylor polynomial expansion is selected, any function can be represented by a matrix of coefficients and exponents that define the polynomial itself.

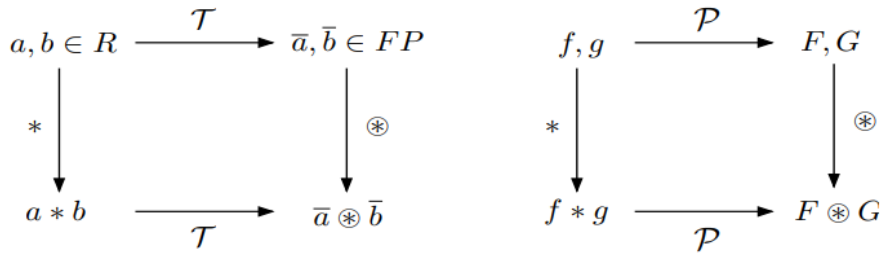

Fig. 1: Analogy between the FP representation of real numbers in a computer environment (left) and the algebra of Taylor polynomials in the DA framework (right) [17,18] .

Hence, algebraic operations in the space of the truncated Taylor series expansion are defined such that they approximate the operations on the function space $\mathcal{C}^r(0)$ of r times differentiable functions at 0. Figure 2 illustrate the process [19,20]. The expression $1/(x+1)$ is evaluated once in $\mathcal{C}^r(0)$ (top) and then in the DA framework with truncation order 3 (bottom). Starting from the identity function x, one is added to get the function $x+1$, the representation of which is fully accurate in the DA framework as it is a polynomial of order 1. The multiplicative inversion is now performed, resulting to the function $1/(1+x)$ in $\mathcal{C}^r(0)$. This function is not a polynomial anymore and it gets automatically approximated in the DA arithmetic by its truncated Taylor expansion around 0, given by $1-x+x^2-x^3$. Note that, by definition of the DA operations, the diagram for each single operation commutes. The same result is reached by first Taylor expanding a $\mathcal{C}^r(0)$ 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 $\mathcal{C}^r(0)$ operation and then Taylor expanding the result [19,20].

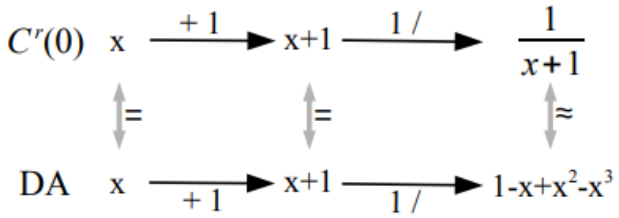

Fig. 2: Evaluation of the expression $1/(1+x)$ in $\mathcal{C}^r(0)$ and in the DA arithmetic [19,20] .

In addition to algebraic operations, the DA framework can be endowed with natural differentiation and integration operators, completing the structure of a differential algebra. For example, the evaluation of the gradient of a function can be performed by working directly on the single monomials of a polynomial, without the need of evaluating derivatives numerically or symbolically.

Of particular interest is the efficient calculation of Taylor polynomial expansions of the flow of ordinary differential equations as a function of its initial conditions. Numerical ODE solvers are based on algebraic operations, involving the evaluation of the derivatives at several integration points. Starting from the DA representation of the initial conditions, $\mathbf{x}_0$, and carrying out all the evaluations in the DA framework, the flow of an ODE is obtained at each step as its Taylor expansion centered at the initial conditions. Given an ODE in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{18}$$

with initial condition $\hat{\mathbf{x}}_0$ at time t_0, the dynamical system can be solved directly in the DA framework. This is achieved by replacing the operations in a classical

integration scheme, including the evaluation of the right hand side, by the corresponding operation in the DA framework . Therefore, after initializing the initial condition in the DA framework

$$\mathbf{x}_0(\delta\mathbf{x}_0) = \hat{\mathbf{x}}_0 + \delta\mathbf{x}_0 \tag{19}$$

where $\delta\mathbf{x}_0$ is the DA variable that represents the deviation form the center of the polynomial, the DA ODE integration allows the propagation of the Taylor series expansion of the flow forward in time, up to any final time t_f. Any explicit ODE integration scheme can be rewritten as a DA integration scheme in a straight forward way. For the particular case regarding this paper, a DA version of the 7/8 Dormand-Prince Runge-Kutta scheme is used. 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 the high order expansions of the flow [21]. As this is achieved by merely replacing algebraic operations on floating-points numbers by DA operations, the method is inherently ODE independent [20]. The results is a polynomial representation of the flow, which can be represented as

$$\mathbf{x}_f(\delta\mathbf{x}_0) = \hat{\mathbf{x}}_f + \mathcal{M}_{0\to f}^{\hat{\mathbf{x}}_0}(\delta\mathbf{x}_0) \tag{20}$$

where $\hat{\mathbf{x}}_f$ indicates the constant part of the polynomial (zeroth order term), which is equivalently calculated propagating the initial condition $\hat{\mathbf{x}}_0$ to the final time t_f. $\mathcal{M}_{0\to f}^{\hat{\mathbf{x}}_0}(\delta\mathbf{x}_0)$ is the (forward) polynomial map and contains all the non-zero-order terms of the expansion. The map is centered at the initial condition $\hat{\mathbf{x}}_0$, and describes how a perturbation at the initial time $\delta\mathbf{x}_0$ maps to the final condition $\mathbf{x}_f(\delta\mathbf{x}_0)$. The map uses the monomial basis in the $\delta\mathbf{x}_0$ variable to describe the flow of the dynamics. The result is an automatic Taylor series expansion of the result of the numerical method, i.e. the numerical approximation of the flow, with respect to any quantity that was initially initialized as a DA variable. Therefore, an efficient implementation of DA allows to obtain high-order expansions with limited computational time [19]. This approach results in potentially replacing thousands of integrations with evaluations of the Taylor expansion of the flow. As a result, in many applications, the computational time reduces considerably without any significant loss in accuracy [17].

For a more complete and detailed explanation on Differential Algebra, please refer to the references. This works uses the Differential Algebra Core Engine (DACE2.0) software tool [18,20].

5 The Differential Algebra Maximum A Posteriori filter (DAMAP)

Given the following discrete time state-space model

$$\mathbf{x}_{k+1} = \phi(\mathbf{x}_k; t_{k+1}, t_k) \tag{21}$$

$$\mathbf{y}_{k+1} = h(\mathbf{x}_{k+1}) + \boldsymbol{\eta}_{k+1} \tag{22}$$

where ϕ is the process model, $\mathbf{x}_k$ is the n-dimensional state vector at time-step k, $\mathbf{y}_{k+1}$ is the m-dimensional measurement vector at time-step $k+1$, and h is the measurements function. The measurement noise $\boldsymbol{\eta}_k$ is a Gaussian random sequence with covariance

$$\mathbb{E}\left\{\boldsymbol{\eta}_i \boldsymbol{\eta}_k^T\right\} = \mathbf{R}\ \delta_{ik} \tag{23}$$

The initialization of the filtering algorithm is performed by assuming a normal prior distribution of the state with mean $\hat{\mathbf{x}}_0$ and covariance $\mathbf{P}_0$.

$$p_{\mathbf{x}_0}(\mathbf{x}_0) = \frac{1}{(2\pi)^{n/2}\sqrt{\det \mathbf{P}_0}} \exp\left(-\frac{1}{2}(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T \mathbf{P}_0^{-1}(\mathbf{x}_0 - \hat{\mathbf{x}}_0)\right) \tag{24}$$

Since Differential Algebra works with deviations from the mean, the DA variable is initialized as

$$\delta\mathbf{x}_0 = \mathbf{x}_0 - \hat{\mathbf{x}}_0 \tag{25}$$

By considering the logarithm of the distribution, the prior can be expressed as a quadratic form.

$$\Xi_{\hat{\mathbf{x}}_0}(\delta\mathbf{x}_0) = -\frac{1}{2}\delta\mathbf{x}_0^{\mathrm{T}}\mathbf{P}_0^{-1}\delta\mathbf{x}_0 \stackrel{+}{=} \log p_{\mathbf{x}_0}(\hat{\mathbf{x}}_0 + \delta\mathbf{x}_0) = \log p_{\mathbf{x}_0}(\mathbf{x}_0) \tag{26}$$

where the symbol $\stackrel{+}{=}$ is introduced to indicate equality modulo an additive constant. Notice that setting the order of the Taylor polynomial expansion of the flow's ϕ and the measurement h to c, results in a representation of the distribution $\Xi_{\hat{\mathbf{x}}_0}(\delta\mathbf{x})$ being a Taylor polynomial of order $2c$ centered at $\hat{\mathbf{x}}_0$.

5.1 Time Propagation

The prediction part of the filter starts with the time propagation of the distribution. Assume that the state evolution is Hamiltonian

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}) \tag{27}$$

then, from the prior results, we have that (dropping the explicit time dependency)

$$p_{\mathbf{x}_{k+1}}(\mathbf{x}_{k+1}) = p_{\mathbf{x}_k}(\boldsymbol{\phi}^{-1}(\mathbf{x}_k; t_{k+1}, t_k)) \tag{28}$$

The aim is to get the polynomial approximation of the inverse of the flux through DA techniques. Suppose that the integration of $\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x})$ from t_k to t_{k+1} is performed using a polynomial approach, a c-th order Taylor polynomials, and that the initial condition $\mathbf{x}_k^+$ (the superscript $+$ indicates the state is from the posterior distribution) is given by a deterministic center $\hat{\mathbf{x}}_k^+$ and a deviation $\delta\mathbf{x}_k$

$$\mathbf{x}_k^+ = \hat{\mathbf{x}}_k^+ + \delta\mathbf{x}_k \tag{29}$$

Under these assumptions, the propagated state using DA is given by

$$\mathbf{x}_{k+1}^{-} = \hat{\mathbf{x}}_{k+1}^{-} + \mathcal{M}_{(k\to k+1)}^{\hat{\mathbf{x}}_k^+}(\delta\mathbf{x}_k) \tag{30}$$

where

$$\hat{\mathbf{x}}_{k+1}^{-} = \phi(\hat{\mathbf{x}}_k^+; t_{k+1}, t_k) \tag{31}$$

The map can be inverted using an algorithm in the DACE2.0 library based on fixed point iterations [22] to obtain

$$\mathcal{W}_{(k+1\to k)}^{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1}) = \left(\mathcal{M}_{(k\to k+1)}^{\hat{\mathbf{x}}_k^+}(\delta\mathbf{x}_k)\right)^{-1} \tag{32}$$

using the properties of Hamiltonian systems, Equation (16),

$$p_{\mathbf{x}_{k+1}}(\mathbf{x}_{k+1}) = p_{\mathbf{x}_{k+1}}(\hat{\mathbf{x}}_{k+1}^- + \delta\mathbf{x}_{k+1}) = p_{\mathbf{x}_k}\left(\hat{\mathbf{x}}_k^+ + \mathcal{W}_{(k+1\to k)}^{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1})\right) \tag{33}$$

Taking logarithms and ignoring constant terms

$$\Xi_{\hat{\mathbf{x}}_{k+1}^-}^{-}(\delta\mathbf{x}_{k+1}) \stackrel{+}{=} \log p_{\mathbf{x}_{k+1}}(\hat{\mathbf{x}}_{k+1}^- + \delta\mathbf{x}_{k+1}) \tag{34}$$

$$= \log p_{\mathbf{x}_k}(\hat{\mathbf{x}}_k^+ + \mathcal{W}_{(k+1\to k)}^{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1})) \tag{35}$$

$$\stackrel{+}{=} \Xi_{\hat{\mathbf{x}}_k^+}^{+}\left(\mathcal{W}_{(k+1\to k)}^{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1})\right) \tag{36}$$

where $\Xi_{\hat{\mathbf{x}}_k^+}^{+}(\delta\mathbf{x}_k)$ is the Taylor polynomial approximation of the logarithm of the posterior distribution of $\mathbf{x}_k$ (modulo a constant) centered at $\hat{\mathbf{x}}_k^+$ and $\Xi_{\hat{\mathbf{x}}_{k+1}^-}^{-}(\delta\mathbf{x}_{k+1})$ is the Taylor polynomial approximation of the logarithm of the prior distribution of $\mathbf{x}_{k+1}$ (modulo a constant) centered at $\hat{\mathbf{x}}_{k+1}^-$.

5.2 Measurements Update

We assume that the measurement $\mathbf{y}_k$ is a (possibly) nonlinear function of the state $\mathbf{x}_k$ corrupted by zero-mean, additive Gaussian noise $\boldsymbol{\eta}_k \sim \mathcal{N}(\boldsymbol{\eta}_k; \mathbf{0}, \mathbf{R}_k)$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \boldsymbol{\eta}_k \tag{37}$$

The Likelihood is defined as

$$L(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}) = \mathcal{N}\big(\mathbf{y}_{k+1}; \mathbf{h}(\mathbf{x}_{k+1}), \mathbf{R}_{k+1}\big) \tag{38}$$

The state is reinitialized as a DA

$$\mathbf{x}_{k+1}^{-} = \hat{\mathbf{x}}_{k+1}^{-} + \delta\mathbf{x}_{k+1} \tag{39}$$

and the measurements are evaluated in the DA framework

$$\mathbf{y}_{k+1} = \mathbf{h}_{k+1}(\mathbf{x}^-_{k+1}) = \bar{\mathbf{y}}_{k+1} + \mathcal{M}^{\hat{\mathbf{x}}^-_{k+1}}(\delta\mathbf{x}_{k+1}) \tag{40}$$

We then achieve the Taylor approximation of the log-likelihood (order $2c$)

$$\begin{aligned}\Lambda_{\hat{\mathbf{x}}^-_{k+1}}(\delta\mathbf{x}_{k+1}) &= -\frac{1}{2}(\mathbf{y}_{k+1} - \bar{\mathbf{y}}_{k+1} - \mathcal{M}^{\hat{\mathbf{x}}^-_{k+1}}(\delta\mathbf{x}_{k+1}))^{\mathrm{T}}\mathbf{R}^{-1}_{k+1}(\mathbf{y}_{k+1} - \bar{\mathbf{y}}_{k+1} - \mathcal{M}^{\hat{\mathbf{x}}^-_{k+1}}(\delta\mathbf{x}_{k+1})) \\ &\stackrel{+}{=} \log L(\mathbf{y}_{k+1}|\mathbf{x}^-_{k+1}) \end{aligned} \tag{41}$$

where, again, the zeroth order term of the polynomial is neglected. The filter could operate even with the introduction of non-Gaussian measurement noise, as long as the noise's PDF is known and its logarithm taken. The MAP estimate is obtained from the maximum of the following function

$$\Xi^+_{\hat{\mathbf{x}}^-_{k+1}}(\delta\mathbf{x}_{k+1}) = \Xi^-_{\hat{\mathbf{x}}^-_{k+1}}(\delta\mathbf{x}_{k+1}) + \Lambda_{\hat{\mathbf{x}}^-_{k+1}}(\delta\mathbf{x}_{k+1}) \tag{42}$$

$$\stackrel{+}{=} \log p_{\mathbf{x}_{k+1}|\mathbf{Y}_{k+1}}(\mathbf{x}^-_{k+1}|\mathbf{Y}_{k+1}) \tag{43}$$

where the notation $\mathbf{Y}_{k+1}$ indicates all measurements $\mathbf{y}_j$ up to and including $j = k+1$. $\Xi^+_{\hat{\mathbf{x}}^-_{k+1}}(\delta\mathbf{x}_{k+1})$ is the Taylor polynomial approximation of the logarithm of the posterior distribution of $\mathbf{x}_{k+1}$ (modulo a constant) centered at $\hat{\mathbf{x}}^-_{k+1}$.

Working directly on the logarithm of PDFs improves the convergence of the polynomials [23] and reduces the computational cost from the machine: a large amount of multiplications is substituted by simpler and fewer additions.

5.3 Maximization

In order to find the MAP estimate, we need to find the deviation that maximizes the posterior distribution:

$$\delta\bar{\mathbf{x}}_{k+1} = \arg\max_{\delta\mathbf{x}_{k+1}} \ \Xi^+_{\hat{\mathbf{x}}^-_{k+1}}(\delta\mathbf{x}_{k+1}) \tag{44}$$

Maximization in the DA framework is performed at first by searching for the zero of the gradient of the posterior obtained using DA as

$$\Theta(\delta\mathbf{x}_{k+1}) = \nabla\Xi^+_{\hat{\mathbf{x}}^-_{k+1}}(\delta\mathbf{x}_{k+1}) \tag{45}$$

This operation is easily performed in the DA framework by working directly with the coefficients and the exponents of the polynomial representation of the distribution. The polynomial approximation of functions leads to fast computations of derivatives, where the gradient (and Jacobian) are easily calculated from monomial differentiation rules. The next step is to find the value of $\delta\mathbf{x}_{k+1}$ that results in a zero gradient. Any algorithm to find the zeros of a nonlinear function can be used, such as the Newton-Raphson method. The initial guess

is set to $\delta\mathbf{x}_{k+1}^{[0]} = \mathbf{0}$ and at each iteration j the Jacobian $\mathcal{J}$ of the gradient is evaluated at $\delta\mathbf{x}_{k+1}^{[j]}$, the next iteration's value is found as

$$\delta\mathbf{x}_{k+1}^{[j+1]} = \delta\mathbf{x}_{k+1}^{[j]} - \mathcal{J}^{-1}\Theta(\delta\mathbf{x}_{k+1}^{[j]}) \tag{46}$$

and the iterations are stopped when a predetermined exit threshold is reached. The deviation that maximizes the posterior distribution is therefore evaluated as the output of the Newton's method. Rather than perform many iterations of the Newton-Raphson algorithm, it is convenient to keep the exit threshold somewhat higher and conclude the maximization in the DA framework. The output of the Newton-Raphson algorithm is used as an accurate initial guess for the DA maximization.

Denote with $\Theta^{+}_{\hat{\mathbf{x}}^{-}_{k+1}}(\delta\mathbf{x}_{k+1})$ the DA representation of the gradient of $\Xi^{+}_{\hat{\mathbf{x}}^{-}_{k+1}}(\delta\mathbf{x}_{k+1})$. This DA gradient is a polynomial and we will express it as the sum of its constant part, $\bar{\Theta}$, and the polynomial map $\mathcal{M}_{\Theta}(\delta\mathbf{x}_{k+1})$, such that

$$\Theta^{+}_{\hat{\mathbf{x}}^{-}_{k+1}}(\delta\mathbf{x}_{k+1}) = \nabla\Xi^{+}_{\hat{\mathbf{x}}^{-}_{k+1}}(\delta\mathbf{x}_{k+1}) = \bar{\Theta} + \mathcal{M}_{\Theta}^{\hat{\mathbf{x}}^{-}_{k+1}}(\delta\mathbf{x}_{k+1}) \tag{47}$$

The polynomial map is inverted

$$\mathcal{W}_{\Theta}^{\hat{\mathbf{x}}^{-}_{k+1}}(\delta\mathbf{x}_{k+1}) = \left(\mathcal{M}_{\Theta}^{\hat{\mathbf{x}}^{-}_{k+1}}(\delta\mathbf{x}_{k+1})\right)^{-1} \tag{48}$$

where both $\mathcal{W}_{\Theta}^{\hat{\mathbf{x}}^{-}_{k+1}}(\delta\mathbf{x}_{k+1})$ and $\mathcal{M}_{\Theta}^{\hat{\mathbf{x}}^{-}_{k+1}}(\delta\mathbf{x}_{k+1})$ are a function of the same variable $\delta\mathbf{x}_{k+1}$. The deviation $\delta\bar{\mathbf{x}}_{k+1}$ that results in a zero gradient of $\Xi^{+}_{\hat{\mathbf{x}}^{-}_{k+1}}(\delta\mathbf{x}_{k+1})$ is obtained evaluating the inverse map at $-\bar{\Theta}$

$$\delta\bar{\mathbf{x}}_{k+1} = \mathcal{W}_{\Theta}(-\bar{\Theta}) \tag{49}$$

Hence, $\delta\bar{\mathbf{x}}_{k+1}$ is the actual numerical value of variable $\delta\mathbf{x}_{k+1}$, which indicates the displacement between the maximum and the center of the series. The sufficient condition for maximization is verified by further deriving the gradient and evaluating the derivative value at $\delta\bar{\mathbf{x}}_{k+1}$.

Therefore, the MAP estimate is given by

$$\hat{\mathbf{x}}^{+}_{k+1} = \hat{\mathbf{x}}^{-}_{k+1} + \delta\bar{\mathbf{x}}_{k+1} \tag{50}$$

All is left to start the next time propagation is to shift the center of the Taylor polynomial from the prior estimate to the posterior estimate of the state, i.e. defining a shift

$$\delta\mathbf{x}^{+}_{k+1} = \delta\bar{\mathbf{x}}_{k+1} + \delta\mathbf{x}_{k+1} \tag{51}$$

Equation (42) is shifted through a simple polynomial evaluation

$$\Xi^{+}_{\hat{\mathbf{x}}^{+}_{k+1}}(\delta\mathbf{x}_{k+1}) = \Xi^{+}_{\hat{\mathbf{x}}^{-}_{k+1}}(\delta\mathbf{x}^{+}_{k+1}) \tag{52}$$

where $\Xi^{+}_{\hat{\mathbf{x}}^{+}_{k+1}}(\delta\mathbf{x}_{k+1})$ is the Taylor polynomial approximation of the logarithm of the posterior distribution (modulo a constant) centered at the MAP estimate. The process is then kept iterative by evaluating, again, the gradient of the function at the new center and checking for the necessary and sufficient conditions.

5.4 Mean Square Error Estimation

The three steps above (Time Propagation, Measurement Update, and Maximization) are all is needed to compute the MAP estimate of the state vector, and they are applied recursively in time. The estimate is accompanied by $\Xi^{+}_{\hat{\mathbf{x}}^{+}_{k+1}}(\delta\mathbf{x}_{k+1})$, which expresses the shape of the PDF, but not the scale, hence the actual uncertainty of the estimate is unknown. When desired, it is possible to add an additional step to the algorithm to endow it with a mean square error (MSE) estimate. Since the computation of the MSE is not needed to close the loop, it can be done asynchronously, at a lower rate, or on demand when needed.

The acceptance-rejection method [24] is used to draw samples from an unnormalized PDF. From these samples it is possible to approximate the bias of the estimate (unlike LMMSE, MAP estimation is typically biased) and the MSE.

Samples are randomly generated from an importance distribution and then accepted or rejected based on a selected criteria (Equation (56)). Let start from

$$p_{\mathbf{x}_{k+1}|\mathbf{Y}_{k+1}}(\hat{\mathbf{x}}^{+}_{k+1}+\delta\mathbf{x}_{k+1}) = \mathcal{K}\exp\left(\Xi^{+}_{\hat{\mathbf{x}}^{+}_{k+1}}(\delta\mathbf{x}_{k+1})\right) \tag{53}$$

where $\mathcal{K}$ is the normalizing constant and $\Xi^{+}_{\hat{\mathbf{x}}^{+}_{k+1}}(\delta\mathbf{x}_{k+1})$ is the PDF logarithm modulo a constant. Let $\tilde{g}(\mathbf{x}_{k+1})$ denote the importance PDF such that

$$\tilde{g}(\hat{\mathbf{x}}^{+}_{k+1}+\delta\mathbf{x}_{k+1}) = \mathcal{A}\exp(g(\delta\mathbf{x}_{k+1})) \tag{54}$$

where $g(\delta\mathbf{x}_{k+1})$ is the exponential part of the distribution and $\mathcal{A}$ the corresponding normalizing constant, such that $\tilde{g}(\hat{\mathbf{x}}^{+}_{k+1}+\delta\mathbf{x}_{k+1})$ integrates to one. The ratio $p_{\mathbf{x}_{k+1}|\mathbf{Y}_{k+1}}(\mathbf{x}_{k+1})/\tilde{g}(\mathbf{x}_{k+1})$ must be bounded from above by a constant $\mathcal{C}>0$; defined as

$$\mathcal{C} \geq \sup_{\delta\mathbf{x}_{k+1}}\left(\frac{p_{\mathbf{x}_{k+1}|\mathbf{Y}_{k+1}}(\hat{\mathbf{x}}^{+}_{k+1}+\delta\mathbf{x}_{k+1})}{\tilde{g}(\hat{\mathbf{x}}^{+}_{k+1}+\delta\mathbf{x}_{k+1})}\right) \tag{55}$$

After $\tilde{g}(\delta\mathbf{x}_{k+1})$ has been selected for the random generation of samples, the acceptance-rejection algorithm selects deviations vectors $\delta\mathbf{x}_{k+1}$, distributed as $\Xi^{+}_{\hat{\mathbf{x}}^{+}_{k+1}}(\delta\mathbf{x}_{k+1})$, through the following steps.

1 - Generate random samples $\delta\mathbf{x}^{(i)}_{k+1} = \mathbf{x}^{(i)}_{k+1} - \hat{\mathbf{x}}^{+}_{k+1}$ where $\mathbf{x}^{(i)}$ are distributed according to $\tilde{g}(\mathbf{x}_{k+1})$.

2 - Generate samples $\mathbf{u}^{(i)}$ independent from $\delta\mathbf{x}_{k+1}^{(i)}$ from the uniform distribution unif$(0,1)$.
3 - Check the inequality

$$\log \mathbf{u}^{(i)} \leq \Xi^{+}_{\hat{\mathbf{x}}^{+}_{k+1}}(\delta\mathbf{x}_{k+1}^{(i)}) - g(\delta\mathbf{x}_{k+1}^{(i)}) - \tilde{\mathcal{C}} \tag{56}$$

If the condition is true, then the point $\delta\mathbf{x}_{k+1}^{(i)}$ is accepted as belonging to the distribution $p_{\mathbf{x}_{k+1}|\mathbf{Y}_{k+1}}(\hat{\mathbf{x}}^{-}_{k+1} + \delta\mathbf{x}_{k+1})$; otherwise the point is rejected. Equation (56) provides the logarithmic form of the classic accept-reject inequality [24], where $\tilde{\mathcal{C}}$ indicates the remaining part of constant $\mathcal{C}$ after having taken the logarithm and simplified terms in $\mathcal{A}$ and $\mathcal{K}$. Indeed, by writing the acceptance-rejection condition in its logarithmic form, Equation (56) loses all dependence on the normalizing constant $\mathcal{A}$ and on the (estimable) constant $\mathcal{K}$.

The algorithm is repeated until the desired amount of points, N, is reached. The bias and MSE are evaluated directly from the samples

$$\mathbf{bias} \simeq \frac{1}{N}\sum_{i=1}^{N} \delta\mathbf{x}_{k+1}^{(i)} \tag{57}$$

$$MSE = \mathbb{E}\left\{\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}^{+}_{k+1}\right)\left(\mathbf{x}_{k+1} - \hat{\mathbf{x}}^{+}_{k+1}\right)^{\mathrm{T}}\right\} \simeq \frac{1}{N}\sum_{i=1}^{N} \delta\mathbf{x}_{k+1}^{(i)} \left(\delta\mathbf{x}_{k+1}^{(i)}\right)^{T} \tag{58}$$

The choice of an importance distribution that resembles the desired distribution highly reduces the number of rejected points. Drawing samples from a too-wide distribution leads to the risk of choosing points too far from the expansion center, where the approximation accuracy of the polynomial series decreases. On the other side, a too-narrow distribution may lead to an underestimate of the MSE.

It is important to re-emphasize that the MSE estimation step is for user-information only and has no bearings in the calculation of the MAP and the log-probability Ξ.

6 Numerical Examples

6.1 Range Measurement

The first example proposed is a simple toy problem to highlight the difference between the Minimum Mean Square Error (MMSE) and the Maximum A Posteriori (MAP) estimators.

Consider a range measurement problem without time propagation [25], where the prior is a Gaussian PDF:

$$\mathbf{x} = \mathcal{N}\left(\mathbf{x}; \begin{bmatrix} -3 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}\right) \tag{59}$$

The sensor provides range measurements from the origin

$$y = \|\mathbf{x}\| + \eta$$

where η is Gaussian, zero mean, and with standard deviation of $\sigma = 0.1$. Figure 3 gives a visual representation of the prior and posterior distributions of $\mathbf{x}$ as well as the likelihood function, given $y = 1$.

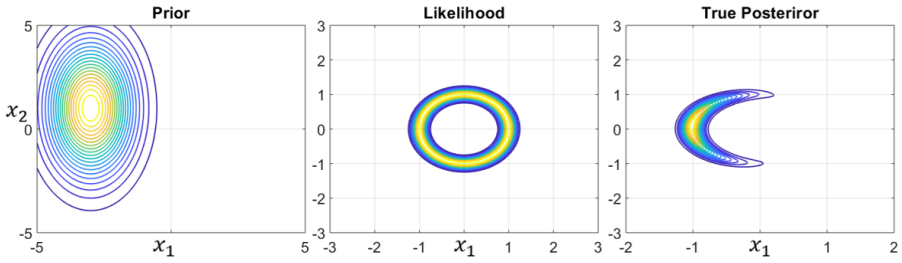

Fig. 3: Prior, likelihood and true posterior probability density functions.

The performance of the Extended Kalman Filter (EKF), Iterated Extended Kalman Filter (IKF) and the Unscented Kalman Filter (UKF) is shown in Fig. 4. Figure 4 shows how these three filters approximate the posterior distribution, parameterized as a Gaussian with mean given by the filter's state estimate and covariance matrix given by the filter's estimate of the estimation error covariance. The EKF and IKF are not producing a consistent estimate,

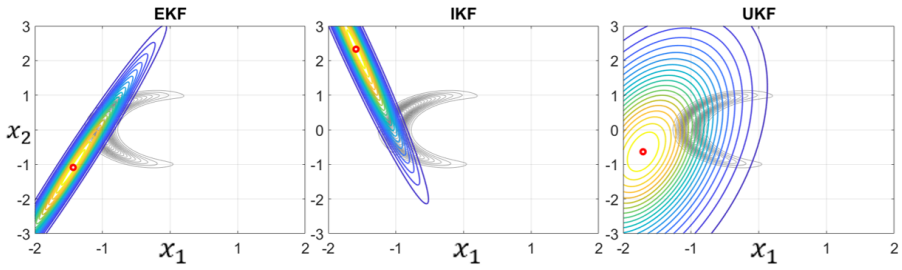

Fig. 4: EKF - IKF - UKF approximation of the posterior PDF.

while the UKF produces an overly conservative and not very accurate, estimate.

Fig. 5 shows DAMAP's approximation of the posterior distribution for different truncation orders c of the Taylor polynomial expansion, achieved by applying the measurement update algorithm (Equation 42).

The polynomial truncation order is added to the name of the filter, DAMAP-c. As an example, DAMAP-2 indicates that the polynomial approximation of

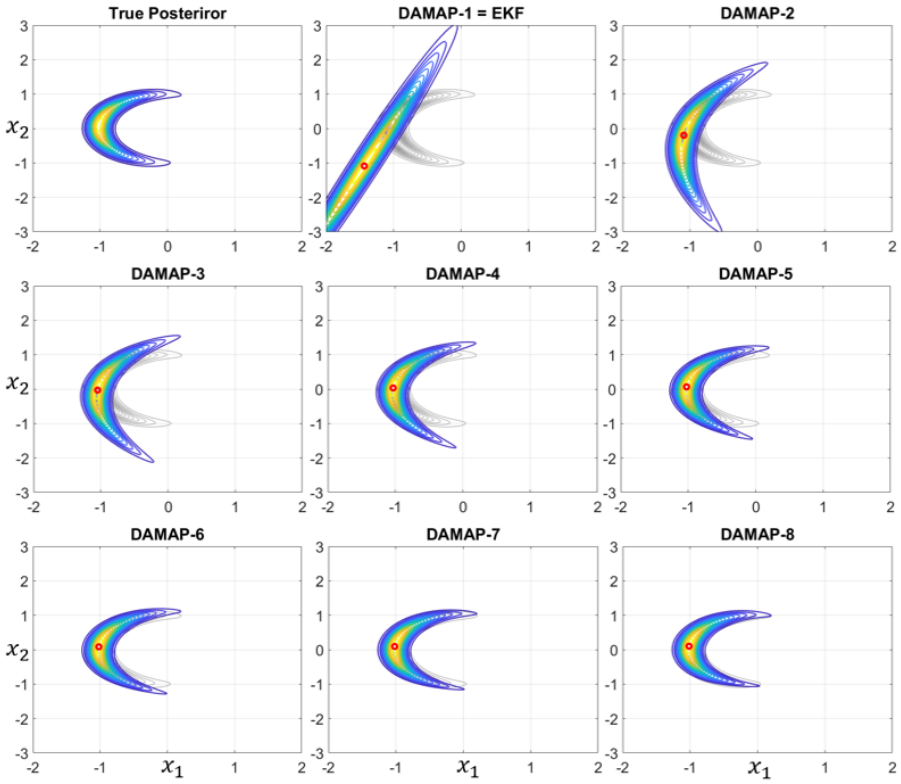

Fig. 5: DAMAP-c approximation of the posterior PDF with different expansion orders.

the state is second order, and consequently the approximation of the log-probability function is 4. As expected, DAMAP-1 behaves exactly like the EKF, being based on the simple linearization of the measurement function. As the order increases, the true PDF approximation improves. DAMAP-8 achieves an excellent representation of the posterior distribution.

Fig. 5 shows the contour lines of the PDF, meaning that there might be a different scaling factor between DAMAP's function and the true one. The scaling factor can be calculated through Monte Carlo integration using importance sampling. Importance samples, $\mathbf{x}^{(i)}$ are not drawn directly from $p(\mathbf{x})$, but from an importance distribution $\pi(\mathbf{x})$, over the same support $\mathcal{S}$. Getting i.i.d (independent and identically distributed) samples from $\pi(\mathbf{x})$

$$\mathbb{E}\left\{f(\mathbf{x})\right\} = \int_{\mathcal{S}} f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \int_{\mathcal{S}} f(\mathbf{x})\frac{p(\mathbf{x})}{\pi(\mathbf{x})}\pi(\mathbf{x})d\mathbf{x} \approx \frac{1}{N}\sum_{i}^{N} f(\mathbf{x}^{(i)})\frac{p(\mathbf{x}^{(i)})}{\pi(\mathbf{x}^{(i)})} \tag{60}$$

which is valid as long as $\pi(\mathbf{x}) \neq 0$ for all values for witch $p(\mathbf{x}) \neq 0$.

To calculate the normalizing scaling constant $\mathcal{K}$, we set $f(\mathbf{x}) = 1$ inside the expectation in Eq. (60) such that the constant is approximated as

$$\frac{1}{\mathcal{K}} = \int \exp\left(\Xi(\mathbf{x})\right) d\mathbf{x} = \int \frac{\exp\left(\Xi(\mathbf{x})\right)}{\pi(\mathbf{x})} \pi(\mathbf{x}) d\mathbf{x} \approx \frac{1}{N} \sum_{i=1}^{N} \frac{\exp\left(\Xi(\tilde{\mathbf{x}}_{(i)})\right)}{\pi(\tilde{\mathbf{x}}_{(i)})} \tag{61}$$

Fig. 6 shows the results of applying importance sampling to the PDF obtained from DAMAP-8. After scaling, the peak of the approximated mode differs only 1.2% from the correct value, proving that the distribution has been correctly normalized. The importance distribution $\pi(\mathbf{x})$ used is Gaussian with mean given by the DAMAP-8 estimate and covariance equal to that of the prior distribution's, Equation (59).

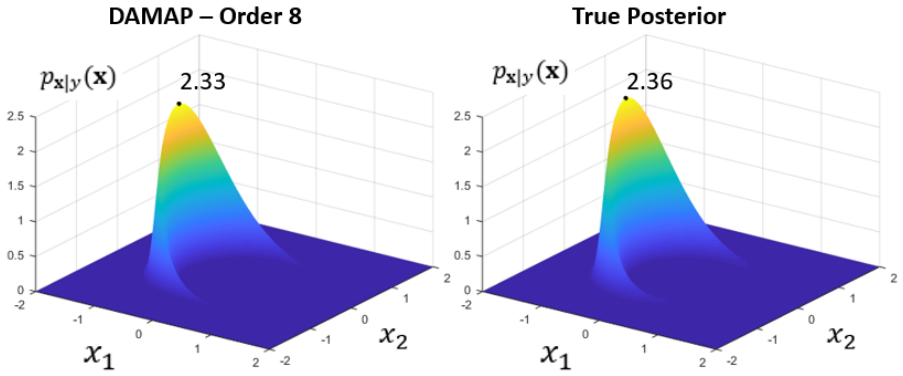

Fig. 6: Mode comparison after importance samplings normalization.

Lastly, DAMAP-c can provide an estimate of the mean square error of the state estimate. In this example, the sampling distribution to calculate the MSE is a uniform distribution centered at the MAP estimate with amplitude 1.5 times the prior PDF's standard deviation. Fig. 7 shows the comparison between the covariance and MSE ellipses between the true distribution and DAMAP-8 approximation. The figure reports in red dots the points that the algorithm accepts for the true distribution, while the blue circles are the ones from the polynomial series. The covariance and the MSE has been evaluated through Monte Carlo calculation according to Equation (58), with 1 million sample points. The figure demonstrate the consistency of the estimation from DAMAP-c and the robustness of the MSE prediction, especially when compared with the covariances of the other filters from Fig. 4. Moreover an estimate of the estimation bias, consisting in the distance between the red point (DAMAP-8 MAP) and the blue point (DAMAP-8 mean), is obtained with Equation (42).

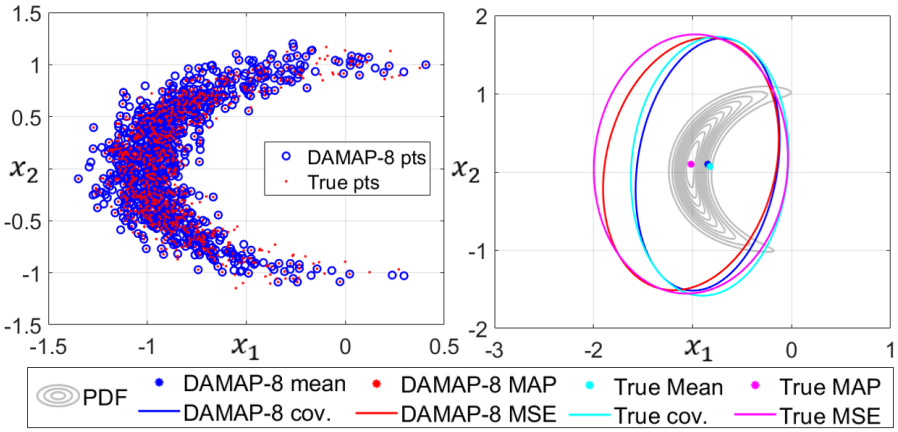

Fig. 7: Covariance and MSE comparison True vs. DAMAP-c.

6.2 Orbit Determination

The proposed filter has been tested on an orbit determination problem. The equations of motion governing the spacecraft are associated to the Keplerian dynamics

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \tag{62}$$

where $\mathbf{r}$ is the position vector and μ is the Earth gravitational parameter. Following the example from [26], the problem has been normalized to be non-dimensional with length scaled by the orbit semi-major axis, $a = 8788$km, and time $\sqrt{\frac{a^3}{\mu}}$. The initial conditions and uncertainties values are taken from previous works [16,19]

$$\mathbf{x}_0 = \begin{pmatrix} \mathbf{r}_0 \\ \mathbf{v}_0 \end{pmatrix} = \begin{pmatrix} -0.68787 \\ -0.39713 \\ 0.28448 \\ -0.51331 \\ 0.98266 \\ 0.37611 \end{pmatrix} \tag{63}$$

The measurements are range and bearing angles and are taken with respect of the center of planet at different acquisition frequencies.

$$y_1 = \|\mathbf{r}\| + \eta_1 \tag{64}$$

$$y_2 = \arctan\left(\frac{x_2}{x_1}\right) + \eta_2 \tag{65}$$

$$y_3 = \arcsin\left(\frac{x_3}{r}\right) + \eta_3 \tag{66}$$

where η_i, $i = 1, 2, 3$, is the measurement noise, assumed to be zero mean, white, and Gaussian. Figure 8 gives a visual representation of the measurements as a function of the state. The standard deviation of the error is assumed to

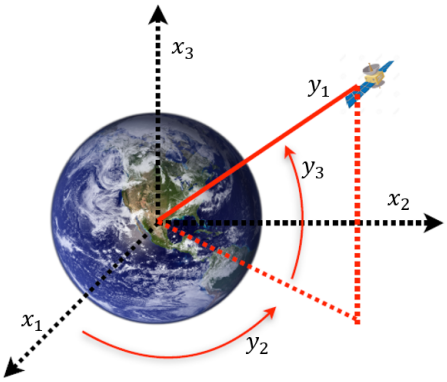

Fig. 8: Visual representation of the measurement equations.

be 0.1m in range and 0.1arcsec for the angles. The initial state uncertainty is assumed to be Gaussian as well, with a diagonal initial error covariance matrix: the value of the variance for the position part of the state vector components is 0.01, while the variance for the velocity components is 10^{-4}. Therefore

$$\sigma_r = 10^{-2} a \tag{67}$$

$$\sigma_v = 10^{-4} \sqrt{\frac{\mu}{a}} \tag{68}$$

A Gaussian PDF with mean $\hat{\mathbf{x}}^+$ and covariance $\mathbf{P}_g$ is chosen as the sampling distribution to calculate the MSE. The covariance $\mathbf{P}_g$ is selected as twice the covariance predicted by a simple EKF update.

$$\tilde{g}(\hat{\mathbf{x}}^+ + \delta\mathbf{x}) = \mathcal{A} \exp(-\frac{1}{2} \delta\mathbf{x}^T \mathbf{P}_g^{-1} \delta\mathbf{x}) \tag{69}$$

The condition from Equation (56) becomes

$$\begin{aligned} \log \mathbf{u}^{(i)} \leq \Xi^+_{\hat{\mathbf{x}}^+_{k+1}}(\delta\mathbf{x}^{(i)}) + \frac{1}{2}(\delta\mathbf{x}^{(i)})^T \mathbf{P}_g^{-1} (\delta\mathbf{x}^{(i)}) + \\ - \sup_{\delta\mathbf{x}} \left(\Xi^+_{\hat{\mathbf{x}}^+_{k+1}}(\delta\mathbf{x}^{(i)}) + \frac{1}{2}(\delta\mathbf{x}^{(i)})^T \mathbf{P}_g^{-1} (\delta\mathbf{x}^{(i)}) \right) \end{aligned} \tag{70}$$

The evaluation of the supremum for each time-step is not an easy task and it requires an accurate, and computational costly, analysis of the state distribution. The assumption that the supremum of the ratio between the two distributions occurs at the current estimate is made and results in

$$\log \mathbf{u}^{(i)} \leq \Xi^+_{\hat{\mathbf{x}}^+_{k+1}}(\delta\mathbf{x}^{(i)}) + \frac{1}{2}(\delta\mathbf{x}^{(i)})^T \mathbf{P}_g^{-1} (\delta\mathbf{x}^{(i)}) - \Xi^+_{\hat{\mathbf{x}}^+_{k+1}}(\mathbf{0}) \tag{71}$$

Looking back at Equation (56), we obtained $\tilde{\mathcal{C}} = \Xi^{+}_{\hat{\mathbf{x}}^{+}_{k+1}}(\mathbf{0})$.

The results of the orbit determination problem solved with DAMAP-3 are reported in Fig. 9 where a Monte Carlo analysis of 1000 test runs has been performed. For each test run, the true initial condition is chosen randomly according to the initial probability density function of the state. The simulation tracks the spacecraft for two full orbits obtaining twelve equally spaced observations for each orbit. The blue lines (effective MSE) represent three times

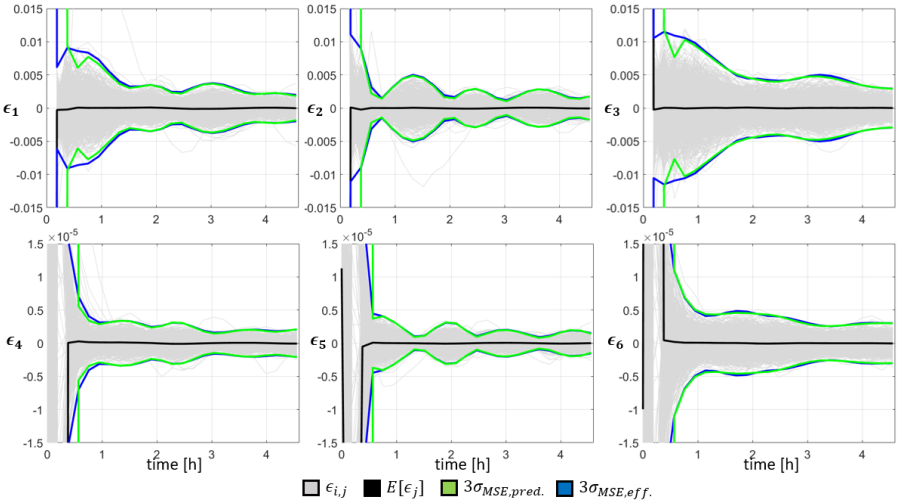

Fig. 9: Monte Carlo analysis of DAMAP-3 with 1000 test runs. Run time of 2 orbits with 12 observation per orbit.

the square root of the mean square error of the 1000 runs. The predicted MSE is calculated using 200 accepted samples and the mean of the 1000 predicted MSEs is shown in green lines (as three times the MSE's square root). The close match between the green and blue lines indicates the filter's consistency.

DAMAP-c is compared to the DA-Based High Order Extended Kalman Filter (EKFDA-k) implemented in previous works [19,26]. This comparison is of importance since it underlines the main differences between a MAP estimator (DAMAP-c) and a Linear Minimum Mean Square Error one. After performing a high-order prediction step using DA techniques, the update step of EKFDA-k resembles that of the classical Kalman filter. Performance of different orders of Taylor polynomial approximations is also investigated. The Unscented Kalman Filter (UKF) is also reported for comparison purposes.

Fig. 10 contains the performance comparison, it shows the MSE profiles for the spacecraft position, left, and velocity, right, on the 2 orbits long simulation with 12 observations per orbit. Each filter compared has two sets of lines associated with it: the dashed lines refer to the standard deviations calculated from the Monte Carlo samples (1000 runs), at each time step, while the continuous

lines are the predicted error standard deviations estimated by each filter. The latter are given by the square root of the sum of the diagonal terms of the updated MSE matrix from the filter. A consistent filter has overlap between its dashed and continuous lines, meaning a match between the effective and the predicted uncertainties.

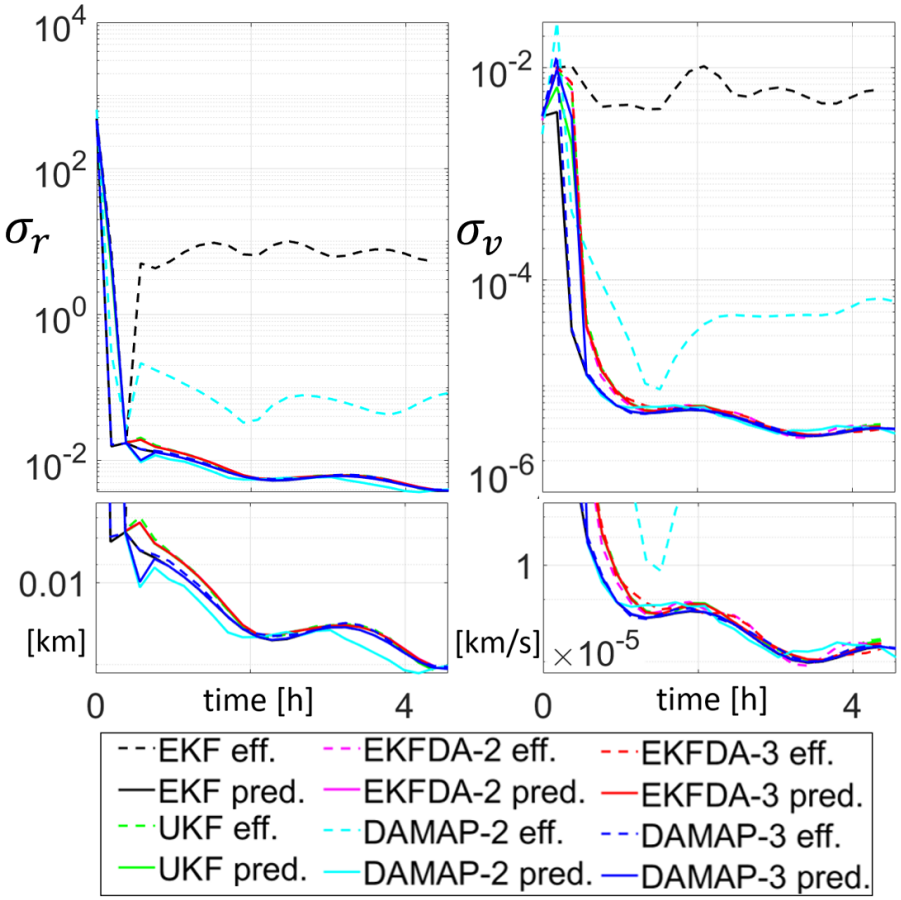

Fig. 10: Comparison of MSE among UKF, DAMAP-c and EKFDA-k for a 2 orbit simulation with 12 observations per orbit.

Fig. 10 shows that the simple linearization of the dynamics does not achieve an accurate estimate and a consistent filter. The EKF (black lines, equivalent to EKFDA-1 and DAMAP-1) diverges and is inconsistent. The UKF, EKFDA-2 and EKFDA-3 behave similarly and achieve similar performance, settling on the same accuracy levels. There appear to be no benefits into increasing the order of EKFDA from two to three: the posterior distribution is not strongly

skewed and third order monomials do not result in an accuracy improvement. DAMAP-3, on the other hand, achieves better results then DAMAP-2: since the second order filter diverges and the dashed cyan line does not overlap with the continuous one. The incoherency is believed to be connected to the nonlinearities and large initial uncertainties. DAMAP-3 converges to a similar accuracy levels as UKF and EKFDA-k. However, the effective MSE shows that DAMAP-3 is the fastest filter during the transient, as shown by the dashed blue line being the lowest among others. The simulation starts at apogee, which explains the increase in the velocity uncertainties after the first step (the overall trace of the MSE matrix is still decreasing).

In order to underline the differences between the above mentioned filters, a second set of simulations is run with the measurement acquisition frequency down to 6 observations per orbit. Fig. 11 shows the Monte Carlo results of DAMAP-3. The matching between the green lines, which represent the estimated MSE, and the blue lines, which are the effective MSE from the 1000 test-runs, states that DAMAP-3 is a consistent filter. DAMAP-3 rapidly reaches steady state and the effective MSE settles to convergence before the estimated MSE. The estimation of the MSE for the first time steps is difficult because the covariance matrix $\mathbf{P}_g$ from the Kalman filter is very large, leading to an overwhelming fraction of rejected samples; therefore, a correct estimation of the uncertainties requires a non feasible amount of time.

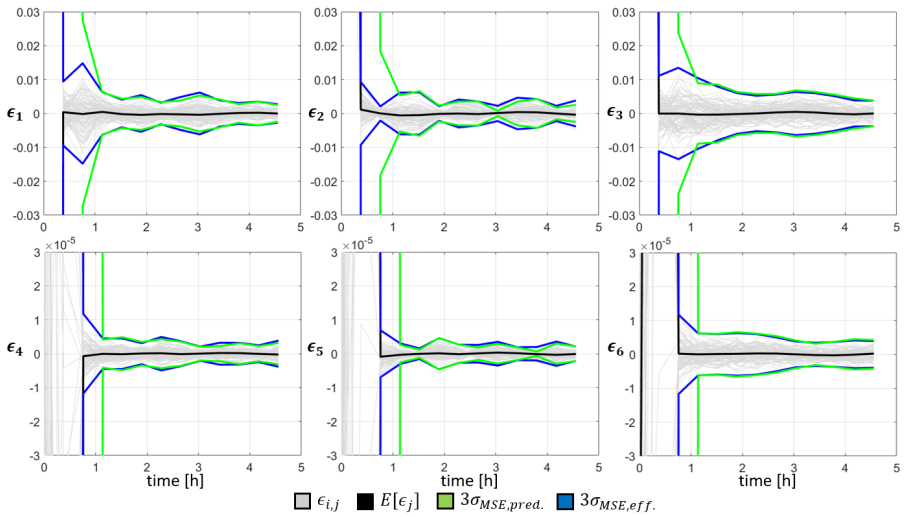

Fig. 11: Monte Carlo analysis of DAMAP-3 with 1000 test runs. Run time of 2 orbits with 6 observation per orbit.

Fig. 12 compares DAMAP-3 with the UKF and EKFDA-3. The UKF, green lines, and EKFDA-3, red lines, show similar results and they have an analogous behavior: the filters achieve convergence but they both underestimate the

MSE level, especially in the transient part of the simulation. The UKF and EKFDA-3 are expected to share a similar pattern in an orbit determination application, as shown in [27,28]. The blue lines represent the performances of DAMAP-3. The figure shows, in a logarithmic scale, that DAMAP-3 is the most consistent filter, with a perfect overlapping of the dashed and continuous lines at steady state. In the transient, the predicted MSE is bigger than the actual error distribution: the acceptance-rejection method is having the same computational issues mentioned previously and it predicts a conservative MSE value that it is larger than the actual one. DAMAP-3 is a nonlinear estimator and it reaches steady state right after the first iteration. The UKF and EKFDA-3 are linear estimators, i.e., the estimate is a linear function of the measurements, and the uncertainties require multiple steps before settling to steady state levels. Therefore, the dashed blue line, which express the actual performance of DAMAP-3, is the lowest among all the others. This difference is marked in the first steps of the simulation, but can also be appreciated by looking at the enlargement of the steady state.

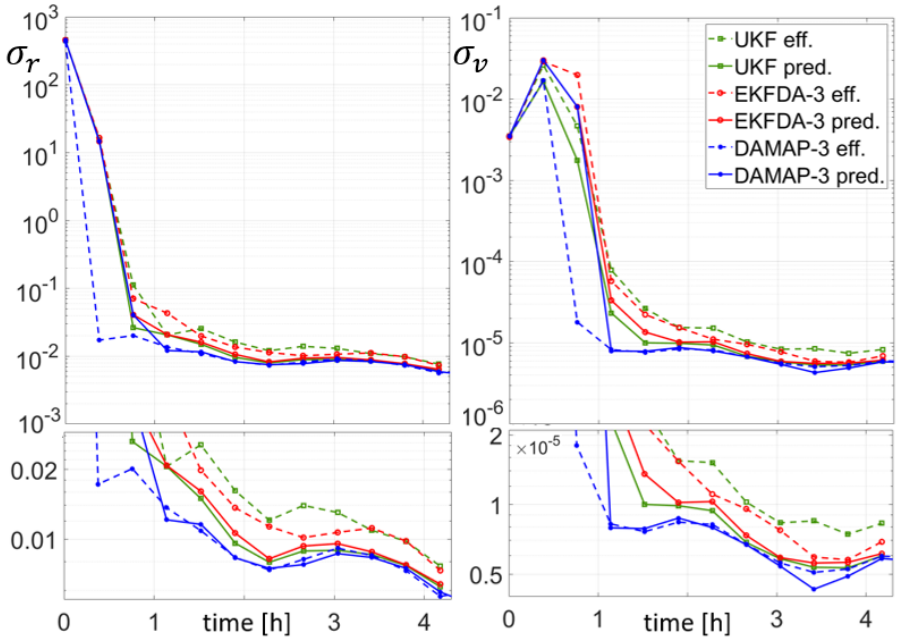

Fig. 12: Comparison of MSE among UKF, DAMAP-3 and EKFDA-3 for a 2 orbit simulation with 6 observations per orbit.

A fully nonlinear MMSE filters will have (by construction) a smaller MSE than a MAP estimator. When compared to linear MMSE estimators, however, DAMAP-3 is able to achieve higher accuracy at steady state due to the precise representation of the posterior distribution through polynomials.

6.3 Orbit Determination - Angles Only

The orbit determination problem is now analyzed when the sensors provide only angular measurements and no range. A Monte Carlo analysis is performed with the same three filters used in the previous simulation. Due to the poor observability, the initial uncertainties (initial state covariance matrix) have been reduced by a factor 4. Fig. 13 shows the 1000 Monte Carlo runs of DAMAP-3 for a 3 orbits long simulation with 24 equally spaced observations per orbit. DAMAP-3 is able to achieve convergence and an accurate estimation of the state of the system is obtained. DAMAP-3 is conservative as the predicted uncertainties are slightly larger than the effective ones from the test runs. The MAP is not necessarily an unbiased estimator, thus DAMAP-c can have a smaller error covariance but a higher MSE than a MMSE estimator.

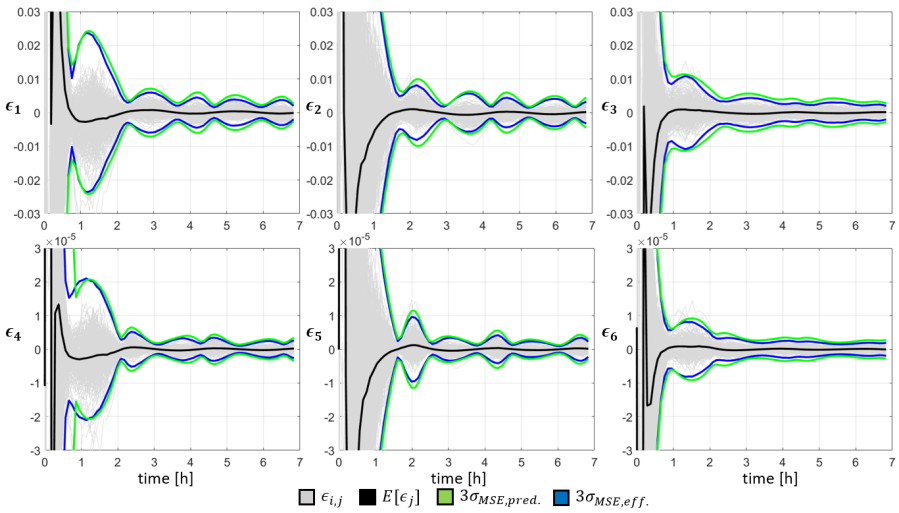

Fig. 13: Monte Carlo analysis of DAMAP-3 with 1000 test runs. Run time of 2 orbits with 24 observations per orbit.

Fig. 14 compares the MSE for position (left) and velocity (right) for various filters. The figure shows, in a logarithmic scale, the steady state behavior of the filters. Therefore, the initial uncertainty levels are out of scale and the figure is zoomed in to enlarge the difference among the different estimators. The figure shows that the UKF is overconfident, the UKF is not able to estimate its own error uncertainties and therefore, even if convergence is achieved, the filter does not operate correctly. The EKFDA-3 is a consistent filter with the best MSE performance. DAMAP performance is worse in the MSE sense than EKFDA-3, which is reasonable since the latter explicitely attempts to minimize the MSE. DAMAP-3 achieves similar accuracy level presented by EKFDA-3, however, it predicts a more conservative uncertainty. The EKF, which consist in the

special case of EKFDA-1, is not reported in the figure because it diverges as in the previous applications.

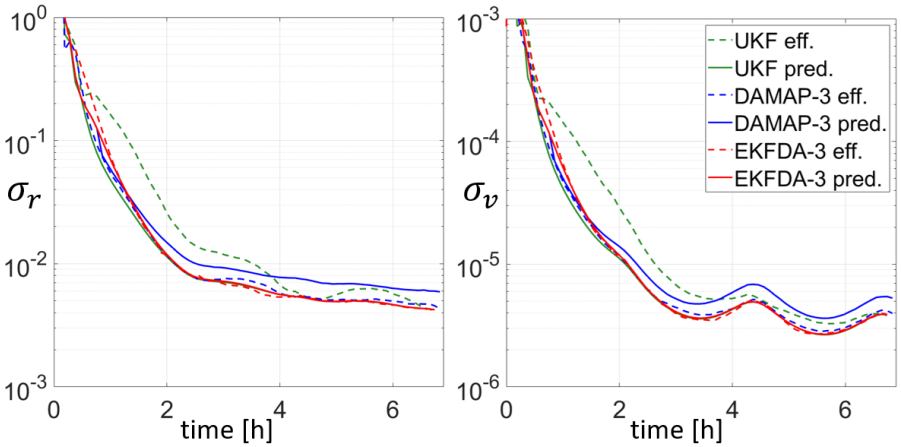

Fig. 14: Comparison of MSE among UKF, DAMAP-3 and EKFDA-3 for a 3 orbit simulation with 24 observations per orbit with angles measurements only. σ_r [km] and σ_v [km/s]

7 Conclusions

A filter based on the Maximum A Posteriori principle has been presented. The capability of propagating and updating the PDF by working directly on the exponent of the distribution had given DAMAP-c the ability to reach satisfying accuracy levels. Thanks to the polynomial representation of the exponent of the PDF, DAMAP-c can accurately estimate the shape of the posterior distribution. If information regarding the scaling factor and uncertainties is required, then DAMAP-c can output both covariance and MSE applying Monte Carlo integration techniques. Moreover, in the simulations performed, due to the higher order polynomial approximation of the exponent of the distribution through DA techniques, the filter is able to reach convergence level faster than high-order filters based on the linear MMSE, such has EKFDA-k and the UKF.

Being based on the MAP principle makes DAMAP-c a nonlinear estimator which is able to achieve better results when compared to linear ones, such as UKF and EKFDA-k. Moreover, DAMAP-c also has the feature to represent the shape of the posterior distribution, rather than simply the covariance matrix.

Acknowledgment

This work was sponsored in part by the Air Force Office of Scientific Research under grant number FA9550-18-1-0351

Appendix

The DAMAP algorithm for the orbit determination application summed up in Algorithm 1.

References

1. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Vol. 82, No. Series D, March 1960, pp. 35–45, doi:10.1115/1.3662552.
2. Kalman, R. E. and Bucy, R. S., "New Results in Linear Filtering and Prediction," *Journal of Basic Engineering*, Vol. 83, No. Series D, March 1961, pp. 95–108, doi:10.1115/1.3658902.
3. Tapley, B. D., Schutz, B. E., and Born, G. H., *Statistical Orbit Determination*, Elsevier Academic Press, 2004, ISBN: 9780080541730.
4. Gelb, A., editor, *Applied Optimal Estimation*, The MIT press, Cambridge, MA, 1974, ISBN:9780262200271.
5. Junkins, J. and Singla, P., "How Nonlinear is it? A tutorial on Nonlinearity of Orbit and Attitude Dynamics," *Journal of the Astronautical Sciences*, Vol. 52, No. 1-2, 2004, pp. 7–60.
6. Julier, S. J. and Uhlmann, J. K., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, 2004, pp. 401–422.
7. Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F., "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, Vol. 45, No. 3, March 2000, pp. 477–482, doi: 10.1109/9.847726.
8. 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.
9. Park, R. S. and Scheeres, D. J., "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.
10. Valli, M., Armellin, R., Di Lizia, P., and Lavagna, M., "Nonlinear mapping of uncertainties in celestial mechanics," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 1, 2012, pp. 48–63, doi: 10.2514/1.58068.
11. Berz, M., *Differential Algebraic Techniques, Entry in Handbook of Accelerator Physics and Engineering*, World Scientific, New York, 1999.
12. Berz, A., "Differential algebraic description of beam dynamics to very high orders," *Part. Accel.*, Vol. 24, No. SSC-152, 1988, pp. 109–124.
13. Hand, L. N.; Finch, J. D., *Analytical Mechanics*, Cambridge University Press, 2008, ISBN 978-0-521-57572-0.
14. Bogachev, V. I., Krylov, N. V., Röckner, M., and Shaposhnikov, S. V., *Fokker-Planck-Kolmogorov Equations*, Vol. 207, American Mathematical Soc., 2015.
15. Witting, A., Lizia, P. D., and R. Armellin et al., "An introduction to Differential Algebra," Dinamica Innovating Technology, European Space Agency.
16. Valli, M., Armellin, R., Di Lizia, P., and Lavagna, M. R., "Nonlinear filtering methods for spacecraft navigation based on differential algebra," *Acta Astronautica*, Vol. 94, No. 1, 2014, pp. 363–374, doi: 10.1016/j.actaastro.2013.03.009.

Algorithm 1 DAMAP

Get $\hat{\mathbf{x}}_0$, $\mathbf{P}_0$, and measurement noise covariance $\mathbf{R}$;
Declare expansion order c, number of samples N, and tolerances τ_1, τ_2;

//Initialization//
$\delta\mathbf{x}_0 = \mathbf{x}_0 - \hat{\mathbf{x}}_0$;
$\Xi_{\hat{\mathbf{x}}_0}(\delta\mathbf{x}_0) = -\frac{1}{2}\delta\mathbf{x}_0^{\mathrm{T}}\mathbf{P}_0^{-1}\delta\mathbf{x}_0$;
while $\tilde{\mathbf{y}}$ is available **do**

//Time Propagation//
$\mathbf{x}_k^+ = \hat{\mathbf{x}}_k^+ + \delta\mathbf{x}_k$; // Initialize Pol.
$\mathbf{x}_{k+1}^- = \mathbf{f}(\mathbf{x}_k^+) = \hat{\mathbf{x}}_{k+1}^- + \mathcal{M}_{(k\to k+1)}^{\hat{\mathbf{x}}_k^+}(\delta\mathbf{x}_k)$; // Pol. Propagation
$\mathcal{W}_{(k+1\to k)}^{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1}) = \left(\mathcal{M}_{(k\to k+1)}^{\hat{\mathbf{x}}_k^+}(\delta\mathbf{x}_k)\right)^{-1}$; // Map Inversion
$\Xi_{\hat{\mathbf{x}}_{k+1}^-}^-(\delta\mathbf{x}_{k+1}) = \Xi_{\hat{\mathbf{x}}_k^+}^+\left(\mathcal{W}_{(k+1\to k)}^{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1})\right)$; // PDF Propagation

//Measurement Update//
$\mathbf{x}_{k+1}^- = \hat{\mathbf{x}}_{k+1}^- + \delta\mathbf{x}_{k+1}$; // Initialize Pol.
$\mathbf{y}_{k+1} = \mathbf{h}_{k+1}(\mathbf{x}_{k+1}^-) = \bar{\mathbf{y}}_{k+1} + \mathcal{M}^{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1})$; // Measurement Pol.
$\Lambda_{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1}) = -\frac{1}{2}(\tilde{\mathbf{y}}_{k+1} - \bar{\mathbf{y}}_{k+1} - \mathcal{M}^{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1}))^{\mathrm{T}}\mathbf{R}_{k+1}^{-1}(\sim)$; // Likelihood
$\Xi_{\hat{\mathbf{x}}_{k+1}^-}^+(\delta\mathbf{x}_{k+1}) = \Xi_{\hat{\mathbf{x}}_{k+1}^-}^-(\delta\mathbf{x}_{k+1}) + \Lambda_{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1})$; // Posterior PDF

//Maximization//
$\Theta(\delta\mathbf{x}_{k+1}) = \nabla\Xi_{\hat{\mathbf{x}}_{k+1}^-}^+(\delta\mathbf{x}_{k+1})$; // Gradient
$\delta\mathbf{x}_{k+1}^{[0]} = \mathbf{0}$;
while τ_1 is achieved **do**
  $\delta\mathbf{x}_{k+1}^{[j+1]} = \delta\mathbf{x}_{k+1}^{[j]} - \mathcal{J}^{-1}\Theta(\delta\mathbf{x}_{k+1}^{[j]})$ // Newton-Raphson
end while
while τ_2 is achieved **do**
  $\Theta_{\hat{\mathbf{x}}_{k+1}^-}^+(\delta\mathbf{x}_{k+1}) = \nabla\Xi_{\hat{\mathbf{x}}_{k+1}^-}^+(\delta\mathbf{x}_{k+1}) = \bar{\Theta} + \mathcal{M}_{\Theta}^{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1})$;
  $\mathcal{W}_{\Theta}^{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1}) = \left(\mathcal{M}_{\Theta}^{\hat{\mathbf{x}}_{k+1}^-}(\delta\mathbf{x}_{k+1})\right)^{-1}$; // Map Inversion
  $\delta\bar{\mathbf{x}}_{k+1} = \mathcal{W}_{\Theta}(-\bar{\Theta})$; // Deviation of MAX
  $\hat{\mathbf{x}}_{k+1}^+ = \hat{\mathbf{x}}_{k+1}^- + \delta\bar{\mathbf{x}}_{k+1}$; // Updated State Pol.
  $\delta\mathbf{x}_{k+1}^+ = \delta\bar{\mathbf{x}}_{k+1} + \delta\mathbf{x}_{k+1}$; // Updated Deviation
  $\Xi_{\hat{\mathbf{x}}_{k+1}^+}^+(\delta\mathbf{x}_{k+1}) = \Xi_{\hat{\mathbf{x}}_{k+1}^-}^+(\delta\mathbf{x}_{k+1}^+)$; // Shifted Posterior PDF
end while

//MSE Estimation (For OD)//
while $i < N$ **do**
  $\mathbf{x}_{k+1}^{(i)}$ from $\tilde{g}(\hat{\mathbf{x}}^+ + \delta\mathbf{x}) = \mathcal{A}\exp(-\frac{1}{2}\delta\mathbf{x}^T\mathbf{P}_g^{-1}\delta\mathbf{x})$;
  $\mathbf{u}^{(i)}$ from unif$(\mathbf{0}, \mathbf{I})$
  $\delta\mathbf{x}_{k+1}^{(i)} = \mathbf{x}_{k+1}^{(i)} - \hat{\mathbf{x}}_{k+1}^+$;
  if $\log\mathbf{u}^{(i)} \le \Xi_{\hat{\mathbf{x}}_{k+1}^+}^+(\delta\mathbf{x}^{(i)}) + \frac{1}{2}(\delta\mathbf{x}^{(i)})^T\mathbf{P}_g^{-1}(\delta\mathbf{x}^{(i)}) - \Xi_{\hat{\mathbf{x}}_{k+1}^+}^+(\mathbf{0})$ **do**
    accept $\mathbf{x}_{k+1}^{(i)}$;
  end if
end while
$MSE = \frac{1}{N}\sum_{i=1}^N \delta\mathbf{x}_{k+1}^{(i)}\left(\delta\mathbf{x}_{k+1}^{(i)}\right)^T$
end while

17. Armellin, R., Di Lizia, P., 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, 2010, pp. 451–470, doi: 10.1007/s10569-010-9283-5.
18. Rasotto, M., Morselli, A., Wittig, A., Massari, M., Di Lizia, P., Armellin, R., Valles, C., and Ortega, G., "Differential algebra space toolbox for nonlinear uncertainty propagation in space dynamics," 2016.
19. Di Lizia, P., Massari, M., and Cavenago, F., "Assessment of onboard DA state estimation for spacecraft relative navigation," 2017, Final report, ESA .
20. Massari, M., Di Lizia, P., Cavenago, F., and Wittig, A., "Differential Algebra software library with automatic code generation for space embedded applications," *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, 2018, p. 0398, doi: 10.2514/6.2018-0398.
21. Wittig, A., Di Lizia, P., Armellin, R., Makino, K., Bernelli-Zazzera, F., and Berz, M., "Propagation of large uncertainty sets in orbital dynamics by automatic domain splitting," *Celestial Mechanics and Dynamical Astronomy*, Vol. 122, No. 3, 2015, pp. 239–261.
22. Berz, M., *Modern Map Methods in Particle Beam Physics*, Academic Press, Sec 2.3.1, https://bt.pa.msu.edu//pub/papers/AIEP108book/AIEP108book.pdf.
23. Armellin, R. and Di Lizia, P., "Probabilistic optical and radar initial orbit determination," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 1, 2018, pp. 101–118.
24. Flury, B. D., "Acceptance–rejection sampling made easy," *SIAM Review*, Vol. 32, No. 3, 1990, pp. 474–476.
25. Tuggle, K. and Zanetti, R., "Automated splitting Gaussian mixture nonlinear measurement update," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 3, 2018, pp. 725–734.
26. Cavenago, F., Di Lizia, P., Massari, M., Servadio, S., and Wittig, A., "DA-based nonlinear filters for spacecraft relative state estimation," *2018 Space Flight Mechanics Meeting*, 2018, p. 1964, doi: 10.2514/6.2018-1964.
27. Servadio, S. and Zanetti, R., "Recursive Polynomial Minimum Mean-Square Error Estimation with Applications to Orbit Determination," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 5, 2020, pp. 939–954.
28. Servadio, S., "High Order Filters For Relative Pose Estimation Of An Uncooperative Target," 2017, Master Thesis, Politecnico di Milano, Milano, Italy.