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

Differential Algebra Based Multiple Gaussians Particle Filter for Orbit Determination

Simone Servadio · Renato Zanetti

Received: date / Accepted: date

Abstract The nonlinear filtering problem plays a fundamental role in multiple space related applications. This paper offers a new filtering technique that combines Monte Carlo time propagation with a Gaussian mixture model measurement update. Differential Algebra (DA) techniques are used as a tool to reduce the computational effort required by particle filters. Moreover, the use of Expectation Maximization (EM) optimization algorithm leads to a good approximation of the probability density functions. The performance of the new method is assessed in the nonlinear Orbit Determination problem, for the challenging case of low observations frequency, and in the restricted three bodies dynamics.

Simone Servadio, Corresponding author
Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin
Austin, Texas
simo_serva@utexas.edu

Renato Zanetti,
Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin
Austin, Texas
renato@utexas.edu

Dedicated to Dr. David Hull, who taught me Optimal Control Theory - Renato

Communicated by Mauro Pontani

Keywords Differential Algebra · Expectation Maximization · Halo Orbits · Nonlinear Filtering · Orbit Determination

1 Introduction

The filtering problem for nonlinear dynamic systems is an important research area that has attracted considerable interest, especially in space applications: it consists in estimating the state of a nonlinear dynamical system from noisy measurements. For the well-known linear and Gaussian case, the posterior distribution remains Gaussian and the optimal estimate is the posterior mean. The well-known Kalman Filter [16] [17] provides the mechanization to calculate the mean and the covariance of the *a posteriori* probability density function (PDF). However, most problems of interest in aerospace engineering applications, such as orbit determination [27], spacecraft navigation, target tracking, etc., require a reliable filtering method that deals with high nonlinearities. In the presence of nonlinearities, the posterior distribution is necessarily a non-Gaussian PDF that is typically not representable exactly in closed form and needs to be approximated.

The most common approach to estimate a system with nonlinear dynamics/measurements is the Extended Kalman Filter (EKF) [8]. The EKF is widely used for trajectory estimation. The EKF linearizes the equations of motion and the measurements equations around the most current estimate and then applies the Kalman filter update equations to the linearized system. In

its Bayesian interpretation, the EKF approximates the distributions as Gaussian. However, in problems with high nonlinearities, the simple linearization assumption fails to provide an accurate approximation of the dynamics and the filter fails to give a valid estimate [15]. In such cases, a different approach that accounts for the system nonlinearities must be used. The Unscented Kalman Filter (UKF) [13] [14] is based on the unscented transformation and does not rely on linearization around the estimate. Carefully chosen sample points are propagated with the true nonlinear dynamics, leading the UKF to higher consistency when compared to the EKF.

Park and Sheeres [21, 22] developed two nonlinear filters that use state transition tensors (STT) to describe the localized nonlinear motion. The initial uncertainties, mean and covariance matrices, are analytically mapped achieving a better representation than the EKF. Valli et al. [30] reproduced Park and Sheeres' work using Differential Algebra (DA), eliminating the need to evaluate STT.

Gaussian Sum Filters (GSF) represent the states distribution function with multiple Gaussian kernels (known as Gaussian Mixture Model, GMM) and apply the Kalman filter equations for each model. The GSF estimate is the weighted sum of the estimates from each model, based on the measurement likelihood function. Accuracy during the PDF time-update has been improved by changing the weights of each distribution regardless of the availability of measurements [28], or by changing the number of kernels to better address the nonlinearities present in the system [6].

The nonlinear filters mentioned above are more accurate than the EKF, but also more computationally expensive due to the evaluation of STT or to performing multiple high fidelity integrations [12]. High accuracy can also be achieved with sequential Monte Carlo methods [26] (such as the particle filter, PF), or the Monte Carlo Kalman filter (MCKF)[10]. However, even if these filters can provide a good estimate of the statics and uncertainties, performing a large number of particle propagations can be computationally expensive and, as a consequence, they are not necessarily always the best choice for orbit determination problems or on-board navigation applications.

Valli et al. [29] use DA techniques to develop a Monte Carlo Kalman filter that substitutes samples propagation with polynomial evaluations: thus it enhances and speeds up the classical Monte Carlo approach. However, the measurement update in [29] is linear, i.e. the distributions are approximated as a Gaussian distribution and the Kalman update is performed. This approach neglects the information about the shape of the predicted PDF contained in the samples. The filter presented in this paper contains a similar prediction step to that in [29], but it improves the update step with the introduction of multiple Gaussian models.

Raihan and Chakravorty [23] developed a particle Gaussian mixture filter where the kernels are created directly from the propagated samples. The estimator has the usual computational burden that characterize all particle filters: having to propagate the whole set of particles. Ref. [23] uses K-means clustering algorithm to form a Gaussian Mixture from the particles. K-means

is a hard clustering algorithm, and the covariance calculation for each Gaussian component is performed using only the points belonging to that cluster, without any influence from the rest of the ensemble. A different approach to clustering is optimization via the Expectation Maximization algorithm [33]. Expectation Maximization is a type of coordinate descent/ascent optimization often used in clustering applications [9].

This paper introduces a filter, EMDAc-N, that combines the strengths of the above cited filters. EMDAc-N will use DA techniques to solve the prediction problem through DA-based Monte Carlo integrations. By solving ordinary differential equations (ODEs) in the DA framework, the result is not only the integrated state, but also a map of how deviations from the nominal solution evolve in time (represented using Taylor series expansion up to a user-defined order c). In the measurement update portion of the algorithm, the performance of the classical Kalman update is improved with the addition of multiple models. After clustering the predicted PDF into N Gaussians, a GSF updates is performed and the estimate is evaluated through weighted mean of the Gaussian components.

This paper is organized as follows. First a brief introduction on differential algebra is presented with references to a more detailed explanation. Then, the main part of the paper describes the filtering algorithm and how it works in the DA framework, underlining the benefits of the Taylor series representation and the improvements given by a multiple model update. Moreover, the effectiveness of the proposed method is assessed in an orbit determination problem

characterized by low availability of measurements: the filter has been compared to multiple others to demonstrate the benefits of the new algorithm. Furthermore, a second orbit determination example is presented where the spacecraft undergoes the restricted three bodies dynamics around the unstable L2 point. Lastly, conclusions are deduced.

2 Differential Algebra

Differential Algebra techniques are used to obtain the c-th order Taylor expansion of the solution flow of a system of ODEs with respect to a given initial condition. DA relies on solving analytical problems using an algebraic approach [18]. Standard representation of functions in a computer environment is based on the simple evaluation at specific points, working with the classical floating point (FP) representation. DA techniques, on the other hand, exploit the idea that it is possible to extract more information from a function rather than its evaluations. Therefore, DA expresses each function as a matrix of coefficients and exponents that describe the Taylor series approximation of that specific function after a center point is selected. The DA framework is able to operate algebraic operations, including differentiation and integration operators, in the DA structure [24]. Therefore, DA offers another way to work in a computer environment, with endowed composition of functions, function inversions, explicit system solving, etc., similar to the algorithms used in FP arithmetic. DA has been successfully used to compute the time evolution of the state of a dynamic system affected by process noise both in discrete-time applications [25]

and in continuous-time situations [19]. Rasotto at al. quantify the accuracy of DA methods by studying the convergence of the approximation errors, for different expansion orders, comparing the DA propagation results with Monte Carlo [24].

An implementation of such DA computer routines is available in the Differential Algebra Core Engine (DACE2.0) software [7], which has been used to implement the algorithm presented in this paper. For additional explanation of DA, the reader is advised to look through previous works such as [29] and [3].

3 Multiple Models Differential Algebra Ensemble Kalman Filter - EMDA

Consider the following system, where the state evolves according to a discrete-time nonlinear state transition equation. The only information about the system is a set of measurements, related to the state vector, acquired at discrete times

$$\mathbf{x}_{k+1} = \mathbf{f}_k[\mathbf{x}_k] + \boldsymbol{\nu}_k \tag{1}$$

$$\mathbf{y}_{k+1} = \mathbf{h}_{k+1}[\mathbf{x}_{k+1}] + \boldsymbol{\eta}_{k+1} \tag{2}$$

where $\mathbf{f}_k$ is the process model, $\mathbf{x}_k$ is the s-dimensional state at time-step k, $\mathbf{y}_{k+1}$ is the m-dimensional vector of the actual measurement at time-step $k+1$, and $\mathbf{h}_k$ is the measurement function. The process noise $\boldsymbol{\nu}_k$ and the measurement

noise $\boldsymbol{\eta}_k$ are random sequences which satisfy the following conditions $\forall\, i, j > 0$:

$$\mathbb{E}\left\{\boldsymbol{\nu}_i\right\} = \mathbb{E}\left\{\boldsymbol{\eta}_i\right\} = \mathbf{0} \tag{3}$$

$$\mathbb{E}\left\{\boldsymbol{\nu}_i \boldsymbol{\nu}_j^T\right\} = \mathbf{Q}\delta_{ij} \tag{4}$$

$$\mathbb{E}\left\{\boldsymbol{\eta}_i \boldsymbol{\eta}_j^T\right\} = \mathbf{R}\delta_{ij} \tag{5}$$

$$\mathbb{E}\left\{\boldsymbol{\nu}_i \boldsymbol{\eta}_j^T\right\} = \mathbf{0} \tag{6}$$

3.1 Cluster Propagation

The truncation order of the Taylor series can be set to any desired value, it is denoted with c. Let N be the fixed number of Gaussian distributions that will approximate the propagated PDF. Therefore, each Gaussian PDF is described by its own mean $\hat{\mathbf{x}}_i$ and covariance matrix $\mathbf{P}_i$, and has an associated weight μ_i, where $i = 1, \ldots, N$. The weights add to one and the estimate is evaluated as the weighted average of the Gaussian kernels' means. If the initial distribution is exactly Gaussian, at initialization each of the N Gaussian components shares the same mean and covariance values, with equal weight $1/N$.

Let n be the total number of particles used to describe the shape of the distribution. Each filter's iteration starts with sampling from the GMM, in particular each ith Gaussian component generates a number of particles proportional to its weight

$$\sigma_i = n\mu_{i,k}^{+} \in \mathbb{N} \tag{7}$$

where $\mu_{i,k}^{+}$ indicates the updated weight at time step k of the i-th Gaussian. The points are generated after calculating, thorough Cholesky Decomposition, the square root of the covariance matrices.

The propagation is carried out in the DA framework . Differential Algebra creates a map of the dynamics that connects deviations at time steps k to deviations at time step $k+1$. This can be achieved by replacing the classic numerical integration scheme with the corresponding DA operations. Consequently, by working directly on functions, the DA solution of any ODE allows the propagation of the Taylor expansion of the flow forward in time from the given initial condition to any final time [32]. Therefore, it is sufficient to propagate the mean of the samples together with the map and, subsequently, evaluate the polynomial map at each deviation. The result is a computational efficient way to evaluate the predicted PDF [1].

The next step of the algorithm is to calculate the deviation of each single sample from its component's mean: this is achieved through simple vector subtraction. As depicted in Figure 1, the deviation δ_j, $j = 1, \ldots, n$, of each

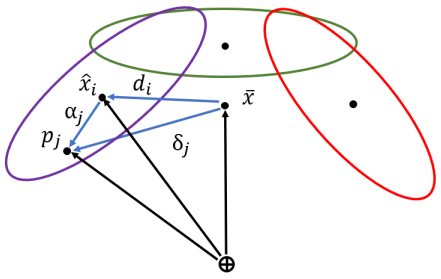

Fig. 1: Vectors representation.

sample from the estimate after propagation $\bar{\mathbf{x}}_k$ is the vector summation of the deviation from the mean of the Gaussian component, α_j, and the distance between that Gaussian mean and the estimate, d_i.

$$\alpha_j = p_{j,k} - \hat{\mathbf{x}}_{i,k} \tag{8}$$

$$d_i = \hat{\mathbf{x}}_{i,k} - \bar{\mathbf{x}}_k \tag{9}$$

$$\delta_j = p_{j,k} - \bar{\mathbf{x}}_k = p_{i,k} - \hat{\mathbf{x}}_{i,k} + \hat{\mathbf{x}}_{i,k} - \bar{\mathbf{x}}_k = \alpha_j + d_i \tag{10}$$

where $p_{j,k}$ is the particle's position and this set of equation is repeated $\forall i, j$.

The state is now propagated to the next time step in the DA framework.

$$\mathbf{x}_k = \bar{\mathbf{x}}_k + \delta\mathbf{x} \tag{11}$$

$$\mathbf{x}_{k+1}^- = \mathbf{f}_k[\mathbf{x}_k] \tag{12}$$

where $\mathbf{x}_{k+1}^-$ indicates the Taylor series expansion of the dynamics centered at $\bar{\mathbf{x}}_k$ and truncated at order c. The polynomial is now evaluated n times, one for each sample point. With this DA approach, $n-1$ propagations are substituted with faster polynomial evaluations. Therefore, each particle of the propagated PDF is found as

$$p_{j,k+1} = \mathbf{x}_{k+1}^-(\delta_j) + \boldsymbol{\nu}_{k+1,j} \quad \forall j = 1, \ldots, n \tag{13}$$

where $\boldsymbol{\nu}_{k+1,j}$ indicates the contribution from the process noise, at time step $k+1$ for the jth particle, generated randomly from the knows process noise PDF.

3.2 K-means

As with all particle filters, the propagated PDF is represented by n random samples. In the proposed approach, the next step is to approximate the distribution as a GMM [23]. After approximating the propagated PDF with multiple Gaussian kernels, the measurements are incorporated as in the Gaussian Sum Filter. Expectation Maximization (EM) is an algorithm that evaluates means and covariances of the Gaussian distributions to approximate the cluster. The EM optimization is initialized with the K-means clustering solution. K-means divides the whole ensemble into a selected number of sets using a hard constraint on the sample: each point is either part of the set or not. EM, on the other hand, enforces a soft constraint where each sample has a probability of belonging to each different set.

Let us randomly select N particles as initial guesses of the means $\hat{\mathbf{x}}^{-}_{i,k+1}$ of the sets for the K-means algorithm. Then, we repeat until convergence the following Equations:

$$a_j := \arg\min_i \|p_{j,k+1} - \hat{\mathbf{x}}^{-}_{i,k+1}\|^2 \tag{14}$$

$$\hat{\mathbf{x}}^{-}_{i,k+1} := \frac{\sum_{j=1}^{n} 1\{a_j = i\} p_{j,k+1}}{\sum_{j=1}^{n} 1\{a_j = i\}} \tag{15}$$

Equation (14) selects which is the closest mean to sample $p_{j,k+1}$ and assigns it to that corresponding set. Equation (15) evaluates the updated means with the new sets of points. When the sets stop changing, the algorithm has reached convergence and the outputs are the N state vectors $\hat{\mathbf{x}}^{-}_{i,k+1}$ and the clustered samples. The K-means algorithm always converges but not necessary to a global optimum. However, since the output of K-means is used purely as a good initial condition for the EM algorithm, the proposed filtering technique does not suffer from this issue. Another possible solution is to implement the more robust K-means++ algorithm [2].

3.3 Expectation Maximization (EM)

As stated above, the EM algorithm is initialized with the output from the K-means algorithm. The initial means are taken from K-means and the initial covariances are calculated directly from the clustered particles. Given the set n independent samples $\rho = \{p_{1,k+1} \ldots, p_{n,k+1}\}$, the goal is to fit the parameters of a model $p(\rho, \zeta)$ to the data, in order to find the maximum of the likelihood

$$\max_{\theta} \ell(\theta) = \max_{\theta} \log \int_{\rho} p(\rho, \zeta; \theta) d\rho \tag{16}$$

where θ are the parameters of the models it is desired to find, such as means, weights and covariances. In Equation (16), the $\zeta_{i,k+1}$ are the latent random variables (unobserved). The EM algorithm gives an efficient method for maximum likelihood estimation. The explicit maximization of $\ell(\theta)$ might be un-

feasible. thus the proposed strategy is to repeatedly construct a lower bound on ℓ, E-step, and then optimize that lower bound, M-step.

Starting from the particles, each one has an associated probability to belong to each of the N Gaussian clusters given by

$$P(p_{j,k+1}|i) = \mathcal{N}\left(\hat{\mathbf{x}}^-_{i,k+1}, \mathbf{P}^-_{i,k+1}\right) = \\ = \frac{1}{(2\pi)^{s/2}\sqrt{\det \mathbf{P}^-_{i,k+1}}} \exp\left(-\frac{1}{2}\left(p_{j,k+1} - \hat{\mathbf{x}}^-_{i,k+1}\right)\left(\mathbf{P}^-_{i,k+1}\right)^{-1}\left(p_{j,k+1} - \hat{\mathbf{x}}^-_{i,k+1}\right)^T\right) \tag{17}$$

where $P(p_{j,k+1}|i)$ is the probability that point $p_{j,k+1}$ belongs to the ith Gaussian. It is now possible to calculate, through Baye's rule, the probability of each Gaussian given the single sample.

$$P(i|p_{j,k+1}) = \frac{P(p_{j,k+1}|i)P(i)}{\sum_{h=1}^{N} P(p_{j,k+1}|h)P(h)} \tag{18}$$

where $P(i) = \mu_{i,k}$ is the weight of each Gaussian component. Due to the exponential behavior of the Gaussian distribution, it is convenient to work with a logarithm scale and to operate with the weights separately. Therefore, the following step of the EM algorithm, after Equation (17), is to calculate the weight of each single particle j referred to each single Gaussian i. This can be performed working directly on logarithms, after some mathematical

manipulations.

$$w_{i,j} = \frac{P(p_{j,k+1}|i)}{\sum_{h=1}^{N} P(p_{j,k+1}|h)} = \frac{1}{\sum_{h=1}^{N} \exp\left(\log\left[P(p_{j,k+1}|h)\right] - \log\left[P(p_{j,k+1}|i)\right]\right)} \qquad \forall i = 1,\ldots,N \wedge \forall j = 1,\ldots,n \tag{19}$$

Conceptually, $w_{i,j}$ defines how much the jth sample belongs to the ith Gaussian: the denominator normalizes the weights such that $\sum_{i=1}^{N} w_{i,j} = 1$. Furthermore, the influence of each Gaussian is evaluated by summing the relative weights.

$$\mathcal{W}_i = \sum_{j=1}^{n} w_{i,j} \tag{20}$$

The new mean and the covariance of the Gaussians can then be calculated:

$$\hat{\mathbf{x}}^{-}_{i,k+1} = \frac{1}{\mathcal{W}_i} \sum_{j=1}^{n} w_{i,j} p_{j,k+1} \tag{21}$$

$$\mathbf{P}^{-}_{i,k+1} = \frac{1}{\mathcal{W}_i} \sum_{j=1}^{n} w_{i,j} \left(p_{j,k+1} - \hat{\mathbf{x}}^{-}_{i,k+1}\right)\left(p_{j,k+1} - \hat{\mathbf{x}}^{-}_{i,k+1}\right)^{T} \tag{22}$$

where $\mathcal{W}_i$ normalizes the summation. The last step is to normalize the weights of the Gaussians such that they sum to unity.

$$\mu^{-}_{i,k+1} = \frac{\mathcal{W}_i}{\sum_{h=1}^{N} \mathcal{W}_h} \tag{23}$$

The next iteration is then ready to start with the new mean, covariance, and weight values until a set tolerance level on the update of the weights is reached.

3.4 Measurement Update

The prediction part of the filter results in the propagated PDF approximated as a GMM. The following step is to perform the measurement update.

The measurement equation can be expressed as a truncated Taylor series expansion in the DA framework.

$$\mathbf{z}_{k+1} = \mathbf{h}_{k+1}[\mathbf{x}_{k+1}] \tag{24}$$

Where $\mathbf{z}_{k+1}$ is a polynomial centered in $\bar{\mathbf{x}}_k$ truncated at order c. The polynomial can now be evaluated using the deviations from Equation (**??**) in order to calculate the predicted measurements associated with each particle.

$$q_{j,k+1} = \mathbf{z}_{k+1}(\tilde{\delta}_j) \tag{25}$$

It is now possible to apply the Kalman filter update equations to each Gaussian component. The measurements covariances and cross covariances are computed directly from the particles:

$$\hat{\mathbf{z}}^-_{i,k+1} = \frac{1}{\mathcal{W}_i} \sum_{j=1}^{n} w_{i,j} q_{j,k+1} \tag{26}$$

$$\mathbf{P}_{ZZ,i} = \frac{1}{\mathcal{W}_i} \sum_{j=1}^{n} w_{i,j} \left(q_{j,k+1} - \hat{\mathbf{z}}^-_{i,k+1}\right) \left(q_{j,k+1} - \hat{\mathbf{z}}^-_{i,k+1}\right)^T + \mathbf{R} \tag{27}$$

$$\mathbf{P}_{XZ,i} = \frac{1}{\mathcal{W}_i} \sum_{j=1}^{n} w_{i,j} \left(p_{j,k+1} - \hat{\mathbf{x}}^-_{i,k+1}\right) \left(q_{j,k+1} - \hat{\mathbf{z}}^-_{i,k+1}\right)^T \tag{28}$$

The covariance matrices $\mathbf{P}_{ZZ,i}$ and $\mathbf{P}_{XZ,i}$ are calculated using the whole ensemble of points. This approach is in contrast with the hard K-means constraint of Ref. [23]. The advantage of a soft constraint is that all Gaussian components are updated using complete knowledge of the distribution from the entire set of points.

The Kalman gain of each Gaussian kernel is evaluated as

$$\mathbf{K}_i = \mathbf{P}_{XZ,i} \left(\mathbf{P}_{ZZ,i}\right)^{-1} \tag{29}$$

Each mean and covariance are updated likewise they are working independently

$$\hat{\mathbf{x}}^+_{i,k+1} = \hat{\mathbf{x}}^-_{i,k+1} + \mathbf{K}_i \left(\mathbf{y}_{k+1} - \hat{\mathbf{z}}^-_{i,k+1}\right) \tag{30}$$

$$\mathbf{P}^+_{i,k+1} = \mathbf{P}^-_{i,k+1} - \mathbf{K}_i \mathbf{P}_{ZZ,i} \mathbf{K}_i^T \tag{31}$$

where $\mathbf{y}_{k+1}$ is the actual measurements vector from the sensors.

3.5 Weights Update and Estimate

The weight associated to each Gaussian component of the updated PDF are updated based on the measurement outcome as well. This step applies Baye's rule to obtain the posterior distribution of the probability of each Gaussian

given the measurements [4].

$$\begin{aligned}\mu^{+}_{i,k+1} &= P(i|\mathbf{Y}_{k+1}) = P(i|\mathbf{y}_{k+1},\mathbf{Y}_k) = \\ &= \frac{P(i,\mathbf{y}_{k+1}|\mathbf{Y}_k)}{P(\mathbf{y}_{k+1}|\mathbf{Y}_k)} = \frac{P(i,\mathbf{y}_{k+1}|\mathbf{Y}_k)}{\sum_{h=1}^{N} P(h,\mathbf{y}_{k+1}|\mathbf{Y}_k)} = \\ &= \frac{P(\mathbf{y}_{k+1}|i,\mathbf{Y}_k)P(i|\mathbf{Y}_k)}{\sum_{h=1}^{N} P(h,\mathbf{y}_{k+1}|\mathbf{Y}_k)}\end{aligned} \tag{32}$$

where $\mathbf{Y}_k$ indicates all the measurements up to time step k. Looking at Equation (32), it can be noted its recursive property, since $P(i|\mathbf{Y}_k) = \mu^{-}_{i,k+1}$. Moreover $P(\mathbf{y}_{k+1}|i,\mathbf{Y}_k)$ is the probability of $\mathbf{y}_{k+1}$ to be the outcome from the ith Gaussian.

$$P(\mathbf{y}_{k+1}|i,\mathbf{Y}_k) = \frac{(2\pi)^{-m/2}}{\sqrt{\det \mathbf{P}_{ZZ,i}}} \exp\left(-\frac{1}{2}\left(\mathbf{y}_{k+1} - \hat{\mathbf{z}}^{-}_{i,k+1}\right)\left(\mathbf{P}_{ZZ,i}\right)^{-1}\left(\mathbf{y}_{k+1} - \hat{\mathbf{z}}^{-}_{i,k+1}\right)^{T}\right) \tag{33}$$

Therefore, the weights update equation can be written as

$$\mu^{+}_{i,k+1} = \frac{P(\mathbf{y}_{k+1}|i,\mathbf{Y}_k)\mu^{-}_{i,k+1}}{\sum_{h=1}^{N} \mu^{-}_{h,k+1} P(\mathbf{y}_{k+1}|h,\mathbf{Y}_k)} \tag{34}$$

These weights will dictate how many samples each Gaussian will generate in the next time step, as explained at the beginning of the algorithm.

Lastly, the state estimate and covariance are calculated as weighted means.

$$\bar{\mathbf{x}}_{k+1} = \sum_{i=1}^{N} \mu^{+}_{i,k+1}\hat{\mathbf{x}}^{+}_{i,k+1} \tag{35}$$

$$\mathbf{P}_{k+1} = -\bar{\mathbf{x}}_{k+1}\bar{\mathbf{x}}^{T}_{k+1} + \sum_{i=1}^{N} \mu^{+}_{i,k+1}\left(\mathbf{P}^{+}_{i,k+1} + \hat{\mathbf{x}}^{+}_{i,k+1}\hat{\mathbf{x}}^{+T}_{i,k+1}\right) \tag{36}$$

The filter is now ready to start the following step with $\bar{\mathbf{x}}_{k+1}$, $\mu^+_{i,k+1}$, $\hat{\mathbf{x}}^+_{i,k+1}$ and $\mathbf{P}^+_{i,k+1}$.

3.6 Algorithm Summary

The proposed filtering technique (EMDAc-N) is summarized in Algorithm 1, with references to Algorithm 2 (K-means), and Algorithm 3 (EM for GMM). Algorithm 3 differs from the classic implementation of the EM algorithm since it adds as outputs the weight of each jth particle $w_{i,j}$, and their sum over the total number of samples, $\mathcal{W}_i$. Due to this feature, Algorithm 1 uses the whole ensemble of points for the evaluation of the covariances $\mathbf{P}_{XZ,i}$ and $\mathbf{P}_{ZZ,i}$, for each ith Gaussian model.

Choosing $N = 1$, EMDAc-1 reduces to the DAEnKF-c from [29]. Adding a linear approximation of the dynamics and measurement equation reduces EMDA1-1 to the classic EKF.

4 Orbit Determination

The performance of the newly developed filter has been assessed with a non-linear problem common in celestial mechanics: the two body problem. The equations of motion governing the system are the ones associated to the Keplerian dynamics, where $\mathbf{r}$ is the position vector of the spacecraft and μ is the

Algorithm 1 EMDAc-N

Declare Taylor truncation order c;
Declare number of particles n;
Declare number of Gaussian Kernels N;
Initialize iteration counter $k = 0$;
Initialize Gaussians: $\mu_{i,0} = 1/N,\ \hat{\mathbf{x}}_{i,0} = \bar{\mathbf{x}}_0,\ \mathbf{P}_i = \mathbf{P}_0\ \forall i = 1, \ldots, N$;
while new measurements $\mathbf{y}_{k+1}$ available **do**

//Deviations//
for $i = 1, \ldots, N$ **do**
$\sigma_i = n\mu^+_{i,k}$; //Number of particles per Gaussian
$p_{j,k} = \mathcal{N}(\mathbf{x}|\hat{\mathbf{x}}_{i,k}, \mathbf{P}_{i,k})$; // Samples generated by each Gaussian
$\delta_j = p_{j,i} - \bar{\mathbf{x}}_k$; //Deviations from current estimate
end for

//Propagation//
$\mathbf{x}_k = \hat{\mathbf{x}} + \delta\mathbf{x}$; //Initialize State Polynomial
$\mathbf{x}^-_{k+1} = \mathbf{f}_k[\mathbf{x}_k]$; //Propagation in the DA framework
for $j = 1, \ldots, n$ **do**
$p_{j,k+1} = \mathbf{x}^-_{k+1}(\delta_j) + \boldsymbol{\nu}_{k+1,j}$; //Propagated ensemble through evaluation
$\tilde{\delta}_j = \delta_j + \boldsymbol{\nu}_{k+1,j}$; //New Deviations with process noise

//Clustering//
$[\hat{\mathbf{x}}^-_{k+1}, \mathbf{P}^-_{k+1}, \boldsymbol{\mu}^-_{k+1}] = \text{K-means}(\mathbf{p}_{k+1}, N)$;
$[\hat{\mathbf{x}}^-_{k+1}, \mathbf{P}^-_{k+1}, \boldsymbol{\mu}^-_{k+1}, \mathbf{w}, \mathcal{W}] = \text{EM}(\mathbf{p}_{k+1}, \hat{\mathbf{x}}^-_{k+1}, \mathbf{P}^-_{k+1}, \boldsymbol{\mu}^-_{k+1})$;

//Measurement Update//
$\mathbf{z}_{k+1} = \mathbf{h}_{k+1}[\mathbf{x}_{k+1}]$; //Evaluate measurement polynomial
for $j = 1, \ldots, n$ **do**
$q_{j,k+1} = \mathbf{z}_{k+1}(\tilde{\delta}_j)$; //Measurement ensemble through evaluation
end for
for $i = 1, \ldots, N$ **do**
$\hat{\mathbf{z}}^-_{i,k+1} = \dfrac{1}{\mathcal{W}_i} \sum_{j=1}^n w_{i,j} q_{j,k+1}$;
$\mathbf{P}_{ZZ,i} = \dfrac{1}{\mathcal{W}_i} \sum_{j=1}^n w_{i,j} \left(q_{j,k+1} - \hat{\mathbf{z}}^-_{i,k+1}\right)\left(q_{j,k+1} - \hat{\mathbf{z}}^-_{i,k+1}\right)^T + \mathbf{R}$;
$\mathbf{P}_{XZ,i} = \dfrac{1}{\mathcal{W}_i} \sum_{j=1}^n w_{i,j} \left(p_{j,k+1} - \hat{\mathbf{x}}^-_{i,k+1}\right)\left(q_{j,k+1} - \hat{\mathbf{z}}^-_{i,k+1}\right)^T$;
$\mathbf{K}_i = \mathbf{P}_{XZ,i} \left(\mathbf{P}_{ZZ,i}\right)^{-1}$;
$\hat{\mathbf{x}}^+_{i,k+1} = \hat{\mathbf{x}}^-_{i,k+1} + \mathbf{K}_i \left(\mathbf{y}_{k+1} - \hat{\mathbf{z}}^-_{i,k+1}\right)$;
$\mathbf{P}^+_{i,k+1} = \mathbf{P}^-_{i,k+1} - \mathbf{K}_i \mathbf{P}_{ZZ,i} \mathbf{K}_i^T$;
$P(\mathbf{y}_{k+1}|i, \mathbf{Y}_k) = \dfrac{(2\pi)^{-m/2}}{\sqrt{\det \mathbf{P}_{ZZ,i}}} \exp\left(-\dfrac{1}{2}\left(\mathbf{y}_{k+1} - \hat{\mathbf{z}}^-_{i,k+1}\right)\left(\mathbf{P}_{ZZ,i}\right)^{-1}\left(\mathbf{y}_{k+1} - \hat{\mathbf{z}}^-_{i,k+1}\right)^T\right)$;
$\mu^+_{i,k+1} = \dfrac{P(\mathbf{y}_{k+1}|i, \mathbf{Y}_k)\mu^-_{i,k+1}}{\sum_{h=1}^N \mu^-_{h,k+1} P(\mathbf{y}_{k+1}|h, \mathbf{Y}_k)}$;
end for

//Estimates//
$\bar{\mathbf{x}}_{k+1} = \sum_{i=1}^N \mu^+_{i,k+1} \hat{\mathbf{x}}^+_{i,k+1}$;
$\mathbf{P}_{k+1} = -\bar{\mathbf{x}}_{k+1}\bar{\mathbf{x}}^T_{k+1} + \sum_{i=1}^N \mu^+_{i,k+1}\left(\mathbf{P}^+_{i,k+1} + \hat{\mathbf{x}}^+_{i,k+1}\hat{\mathbf{x}}^{+T}_{i,k+1}\right)$;
end while

Algorithm 2 K-means

Get ensemble of n points $\mathbf{p}$;
Get number of clusters N;
Declare N random initial means $\hat{\mathbf{x}}_i$;
while $\hat{\mathbf{x}}_i$ changes **do**

//Sample Assigment//
for $j = 1, \ldots, n$ **do**
for $i = 1, \ldots, N$ **do**
$d_i = \|p_j - \hat{\mathbf{x}}_i\|^2$; //Get distances
end for
$a_j = \arg(\min(\mathbf{d}))$; //Assign sample to cluster
end for

//Mean Correction//
for $i = 1, \ldots, N$ **do**
$\hat{\mathbf{x}}_i = \dfrac{\sum_{j=1}^{n} 1\{a_j = i\} p_j}{\sum_{j=1}^{n} 1\{a_j = i\}}$;
end for
end while

Earth gravitational parameter.

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} \tag{37}$$

The initial condition and uncertainties values are chosen equal to those in Refs. [29,7] and are here listed. Length units are normalized by the orbit semi-major axis, $a = 8788\ km$, and time units by the parameter $\sqrt{\dfrac{a^3}{\mu}}$ based

Algorithm 3 EM

Get ensemble of n points $\mathbf{p}$;
Get N means $\hat{\mathbf{x}}_i$, covariances $\mathbf{P}_i$, and weights μ_i;
Declare a tolerance τ;
while tolerance is met **do**

//Maximization//
for $j = 1, \ldots, n$ **do**
for $i = 1, \ldots, N$ **do**

$$P(p_j|i) = \frac{(2\pi)^{-s/2}}{\sqrt{\det \mathbf{P}_i}} \exp\left(-\frac{1}{2}\left(p_j - \hat{\mathbf{x}}_i\right)\left(\mathbf{P}_i\right)^{-1}\left(p_j - \hat{\mathbf{x}}_i\right)^T\right);$$

$$w_{i,j} = \frac{1}{\sum_{h=1}^{N} \exp\left(\log\left[P(p_{j,k+1}|h)\right] - \log\left[P(p_{j,k+1}|i)\right]\right)};$$

end for
end for

//Expectation//
for $i = 1, \ldots, N$ **do**

$$\mathcal{W}_i = \sum_{j=1}^{n} w_{i,j};$$

$$\hat{\mathbf{x}}_i = \frac{1}{\mathcal{W}_i} \sum_{j=1}^{n} w_{i,j} p_j;$$

$$\mathbf{P}_i = \frac{1}{\mathcal{W}_i} \sum_{j=1}^{n} w_{i,j} \left(p_j - \hat{\mathbf{x}}_i\right)\left(p_j - \hat{\mathbf{x}}_i\right)^T;$$

end for
end while

on the orbital period.

$$\mathbf{x}_0 = \begin{pmatrix} \mathbf{r}_0 \\ \mathbf{v}_0 \end{pmatrix} = \begin{pmatrix} -0.68787 \\ -0.39713 \\ 0.28448 \\ -0.51331 \\ 0.98266 \\ 0.37611 \end{pmatrix} \tag{38}$$

and the initial estimate of the system state has a 10% offset from the true initial state.

The measurement model assumes the radial position of the spacecraft w.r.t. the Earth and the line of sight direction of the planet:

$$y_1 = r + \eta_1 \tag{39}$$

$$y_2 = arctan2(\frac{x_2}{x_1}) + \eta_2 \tag{40}$$

$$y_3 = arcsin\Big(\frac{x_3}{r}\Big) + \eta_3 \tag{41}$$

where η_i, with $i = 1, 2, 3$, is the measurement noise, assumed to be Gaussian. The standard deviation of the error is assumed to be 0.1 m for the radial position and 0.1 arcsec for the angle errors. The initial uncertainties are assumed to be Gaussian as well, with a diagonal covariance matrix divided into position states, with std $\sigma_r = 10^{-2}a$, and velocity states, with std $\sigma_v = 10^{-4}\sqrt{\frac{\mu}{a}}$.

For the presented application, the number of Gaussians in the multiple models is chosen beforehand and it is kept fixed during the whole simulation. However, if needed, the EMDAc-N algorithm can be enhanced with merging, splitting, pruning and truncation, according to the methodology that the user desires. The nomenclature EMDAc-N indicates with c the truncation order of the dynamics flow during propagation and N indicates the number of Gaussians used in the clustering algorithm. For example, EMDA2-3 indicates a filter with truncation order 2 and a GMM update with 3 Gaussians. In each single simulation a total of 10^4 particles are used.

A Monte Carlo analysis is performed to assess the consistency of the presented filter and to show reliability and accuracy levels. The simulation is

performed with low acquisition measurement frequency. Figure 2 and Figure 3 show the performance of EMDA2-3 and EMDA2-5, respectively, with a Monte Carlo analysis of 100 runs. The figures show the consistency of the position components, left columns, and velocity components, right columns, of the spacecraft state vector in a simulation with time duration of 12 orbits with 3 equally spaced observations per orbit. Both filters converge and correctly predict the estimation uncertainties. The continuous blue lines indicate the standard deviation of the estimation error as predicted by the filter, expressed as 3σ values, while the dashed blue lines represent the actual standard deviations of the errors calculated directly from the Monte Carlo samples, again shown as 3σ values. The consistency of the filter is assessed by the overlapping of the two lines. Lastly, the black line shows the mean of the samples: the expected value of the error is very close to zero, making EMDAc-N and unbiased filter, matching the theoretical results expected for minimum mean square error (MMSE) estimators.

In order to assess the filter relative accuracy and robustness, it is compared with common estimators, such as the EKF and the UKF, and filters from previous works on differential algebra, such as DAHO-c from [31] and DAEnKF-c from [29]. DAHO-c is a DA-based filter that uses Taylor expansion series up to order c to represent each predicted variable. After propagation in time in the DA framework, the predicted means and covariances are evaluated directly on the monomials of each polynomial using a Gaussian assumption of the distribution. DAEnKF-c is a Monte Carlo Kalman Filter (MCKF) that uses

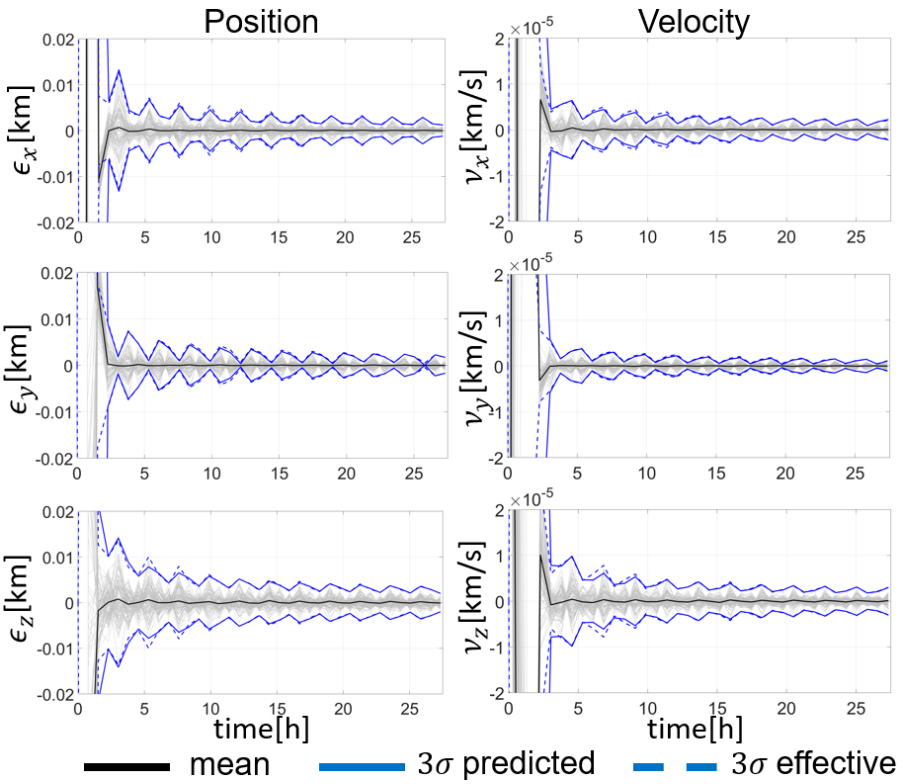

Fig. 2: 100 Monte Carlo runs depicting the performance of EMDA2-3 over 12 orbits with 3 observations each orbit.

polynomial evaluations, through DA, to propagate an ensemble of particles to the next time step, as done in EMDAc-N. Therefore DAEnKF-c is equivalent to choosing a single Gaussian kernel in our approach (EMDAc-1), where the GMM update reduces to the single model Kalman filter update.

Figure 4 is divided into two parts. It shows the standard deviation profiles for the spacecraft position, top row, and velocity, bottom row on a 6 orbits-long simulation with 3 observations per orbit. Each graph has two sets of lines: the dashed lines refer to the standard deviations calculated from the Monte Carlo samples (100 runs), at each time step, while the continuous lines are

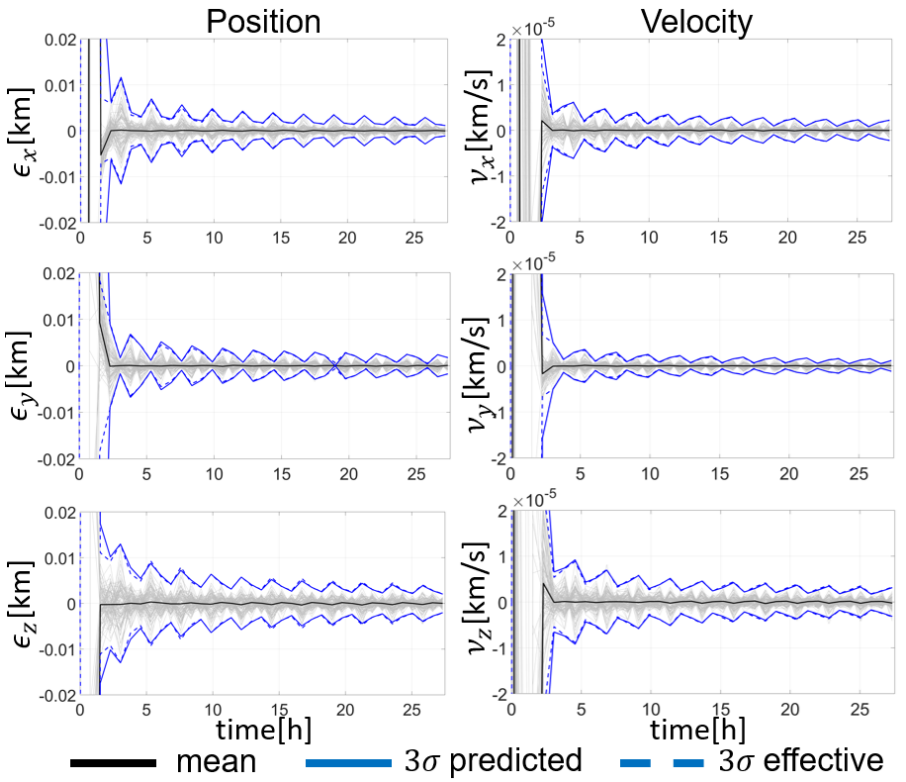

Fig. 3: 100 Monte Carlo runs depicting the performance of EMDA2-5 over 12 orbits with 3 observations each orbit.

the predicted standard deviations estimated by each filter. These values are derived from the diagonal terms of the updated covariance matrix of each of the filters.

$$\sigma_r = \sqrt{\sigma_{rx}^2 + \sigma_{ry}^2 + \sigma_{rz}^2} \tag{42}$$

$$\sigma_v = \sqrt{\sigma_{vx}^2 + \sigma_{vy}^2 + \sigma_{vz}^2} \tag{43}$$

Therefore, a consistent filter will have the overlapping of its dashed and continuous lines, meaning a match between the effective and the predicted uncertainties.

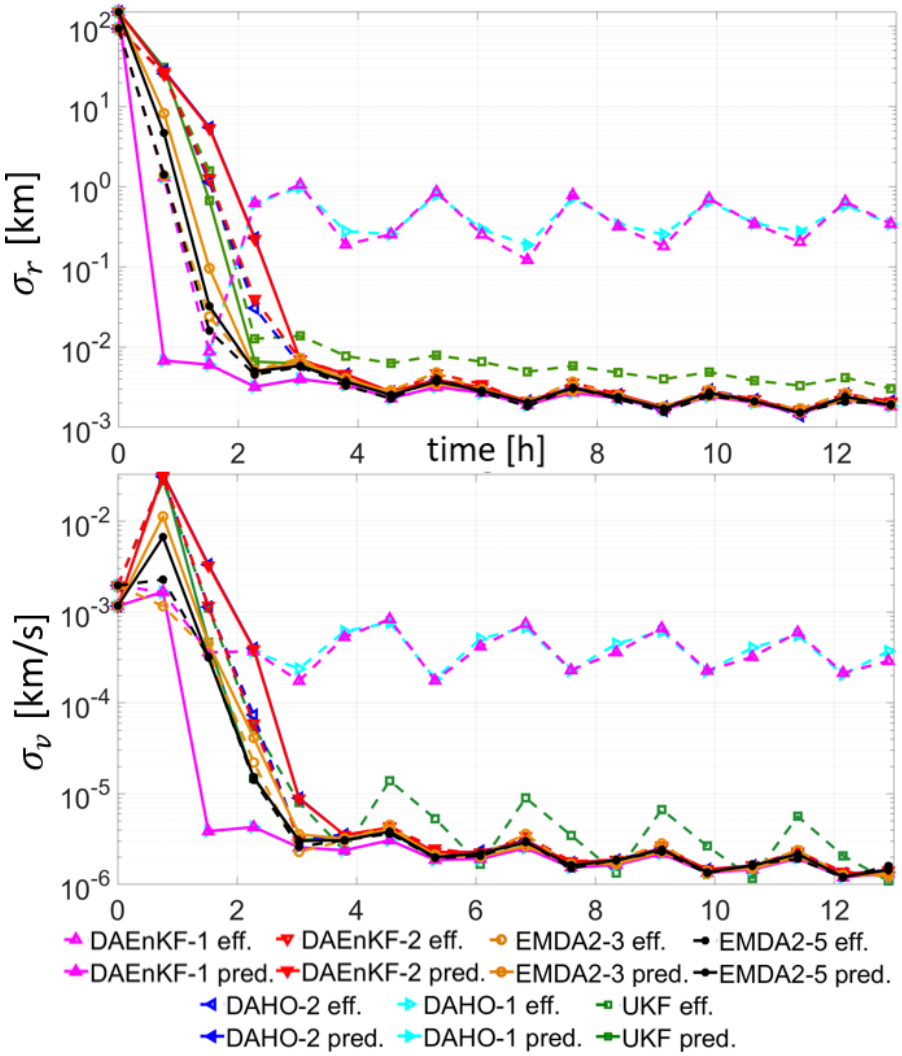

Fig. 4: Position and velocity error standard deviations comparison over a period of 6 orbits with 3 observations per orbit.

The figure compares, in a logarithmic scale, the estimators mentioned above for a total of 7 different filters. From the figure it can be noted how DAHO-1 and DAEnKF-1 have identical trends and they give the same estimation. Their lines overlap both for the predicted and the effective covariance, settling on different order of magnitude. The two filters work with a simple linearization of the dynamics, thus they reduce to the well-know Extended Kalman Filter which, in this orbit determination problem with high initial uncertainty, diverges. In fact, their dashed lines settle two orders of magnitude above the continuous lines: the EKF is overconfident on its estimation and the effective accuracy level is way bigger with respect to the covariance level that the filter is expecting to achieve. The other filters, characterized by nonlinear propagation, behave similarly, but with some important differences that need to be pointed out. The UKF predicted covariance settles with the same accuracy level as the other nonlinear filter, however, the effective covariance from the Monte Carlo runs does not match this prediction and it is larger, indicating the UKF is not performing correctly and it is not a consistent estimator, as the dashed green line in the figure is the only one that does not overlap with the others, both in position and velocity. In contrast, DAHO-2, DAEnKF-2, EMDA2-3 and EMDA2-5 show similar behavior and achieve the same accuracy level. The filters are consistent and the state is predicted correctly with the same steady state precision. However, it is important to point out that the new EMDA2-5 is the fastest filter to reach steady state, as it can be seen by the black lines being the lowest in the transient phase. Furthermore, by

remembering that DAEnKF-2 is equivalent to EMDA2-1, the figure shows that, in EMDA2-N, the bigger the number of Gaussians used in the clustering algorithm, the faster the filter converges to steady state.

The estimation improvement gained by approximating the shape of the propagated PDF with multiple Gaussians can be appreciated by reducing further the measurements acquisition frequency in the simulations. Figure 5 and Figure 6 show the Monte Carlo analysis when only three measurement acquisitions every two revolutions are available, for EMDA2-3 and EMDA2-5 in a 12 orbits long simulation. EMDA2-3 is able to estimate the state of the system, but its covariance prediction is not consistent with the Monte Carlo analysis: the standard deviation calculated from the samples at each time step is slightly bigger when compared to the predicted one, thus the filter diverges. On the other hand, EMDA2-5 has a consistent behavior and both the state and the uncertainties are evaluated correctly, achieving steady state and reaching good accuracy level.

Figure 7 represents the standard deviations analysis for DAHO-2, DAEnKF-2, EMDA2-3, EMDA2-5 and the UFK in a 12 orbits simulation with 3 equally spaced observations each 2 revolutions. The figure resemble the same characteristics from the 3 observations per orbit case. Therefore, each dashed line is connected to a Monte Carlo analysis performed with the relative filter where, at each time step, the standard deviation of the error has been evaluated by extracting the values from the single runs. The continuous lines represent the error standard deviations predicted by the filters according to Equation (42)

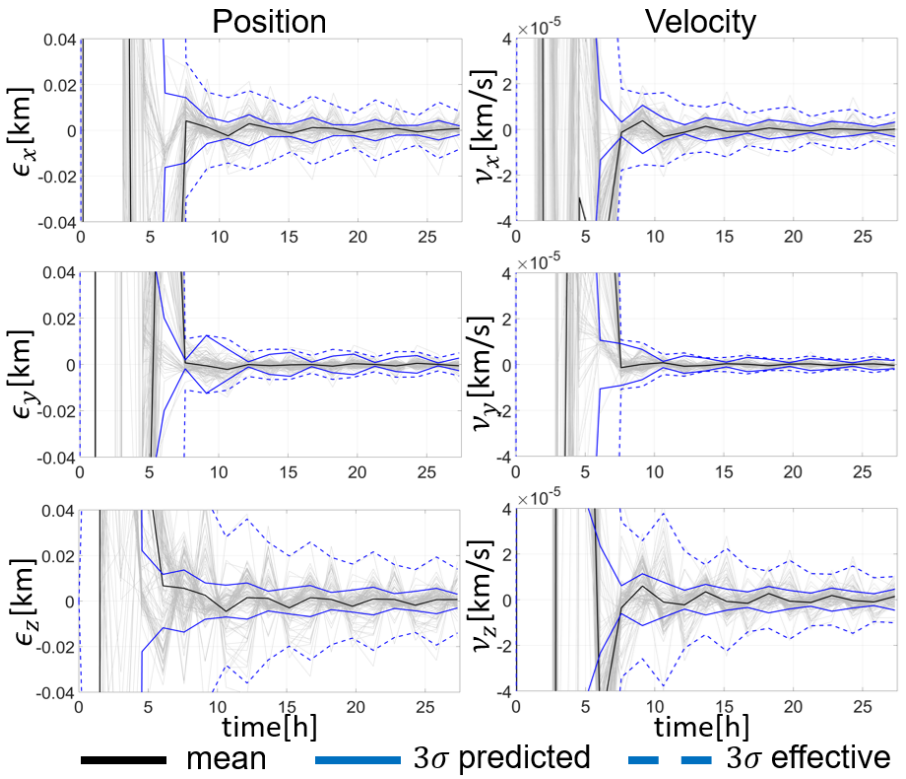

Fig. 5: 100 Monte Carlo runs depicting the performance of EMDA2-3 over 12 orbits with 3 observations each 2 orbits

and (43). The EKF, (equivalent to DAHO-1 and DAEnKF-1) has the same problems seen in Figure 4: it diverges and the effective uncertainties increase, it has therefore been omitted from the figure. The main difference with respect to the previous case comes from UKF, DAHO-2 and DAEnKF-2: they fail in their task to estimate the state and the estimation error grows in time, diverging. Indeed, the dashed blue, red and green lines are almost 9 orders of magnitude out of scale when compare to the relative continuous one. These 3 filters overestimate their confidence in their approximation of the uncertainties and fail to perform a correct update of the state. On the contrary, EMDA*c*-

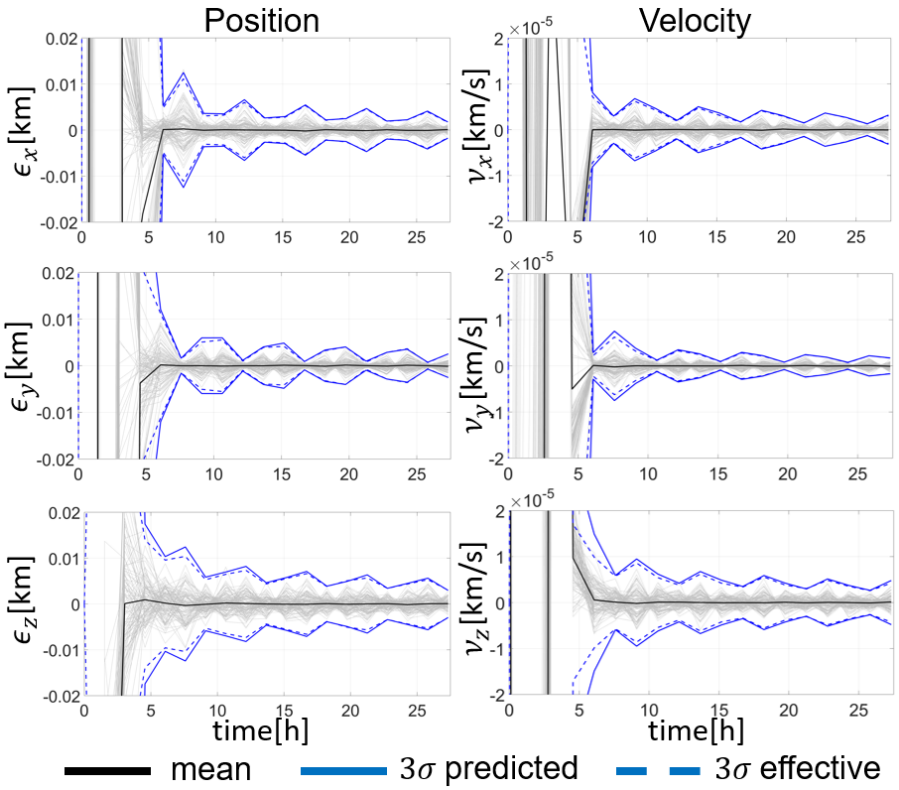

Fig. 6: 100 Monte Carlo runs depicting the performance of EMDA2-5 over 12 orbits with 3 observations each 2 orbits

N is able to estimate the state of the system. When working with multiple Gaussians ($N > 1$), the approximation of the shape of the ensemble of points (thus the distribution) significantly improves and the multiple update achieves a good estimate. However, even if decreasing, the effective std from EMDA2-3 (dashed orange line) does not match the predicted one (continuous orange line), as expected after studying Figure 5: the two lines do not overlap. Otherwise, EMDA2-5 shows robustness and coherence. The black lines demonstrate that EMDA2-5 is the only consistent filter and it has also fast convergence to steady state level. Moreover, during the transient, the effective std stays

below the predicted one. Therefore, it has been proven that the bigger the number of Gaussians used in the algorithm, the more robust the prediction is, at the expenses of a higher computational effort. The Bootstrap PF has not been reported in the figure. In the absence of process noise, the particles of the BPF do not spread during the propagation and the BPF suffers from particle impoverishment.

Figure 8 supports the latter statement by showing the EM clustering at the end of the first propagation. The samples distribution shows how the predicted PDF has the so-called banana shape, characteristic of an orbit determination problem with long propagation times in-between measurements. The clustering algorithm with 5 Gaussians better approximates the shape of the distribution, especially near the mean and at the tails. Using only 3 Gaussians the algorithm does not match the curve of the density function. EMDA2-5 achieves a correct estimate when EMDA2-3 fails.

5 Unstable and Chaotic L2 Orbit

The new filter is also tested on the restricted three body problem. A spacecraft orbiting in the Earth-Moon system has his coordinates expressed with respect to the rotating syndic reference frame. The equations of motion are derived by expressing the position of the spacecraft with respect to the barycenter of the system, which is the center of the frame, as shown by Figure 9. Figure 9 illustrates the x_1 axis connecting Earth's center to the Moon's center and the x_3 axis defining the Moon's rotation plane around the Earth. The x_2 axis

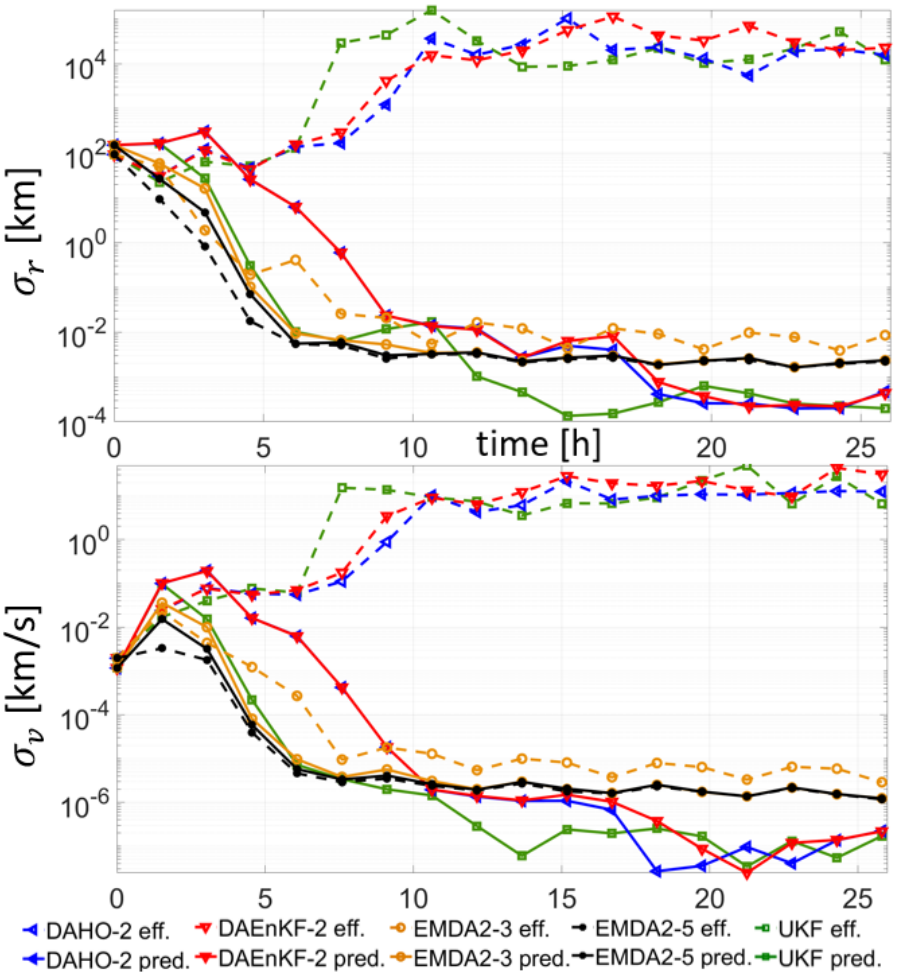

Fig. 7: Position and velocity error standard deviations comparison over a period of 12 orbits with 3 observations each 2 orbits.

closes the frame. The problem is scaled such that variables and units are non dimensional. The location of the barycenter for the system is defined by the mass ratio

$$\mu^* = \frac{m_{MOON}}{m_{MOON} + m_{EARTH}} = 0.0121505856 \tag{44}$$

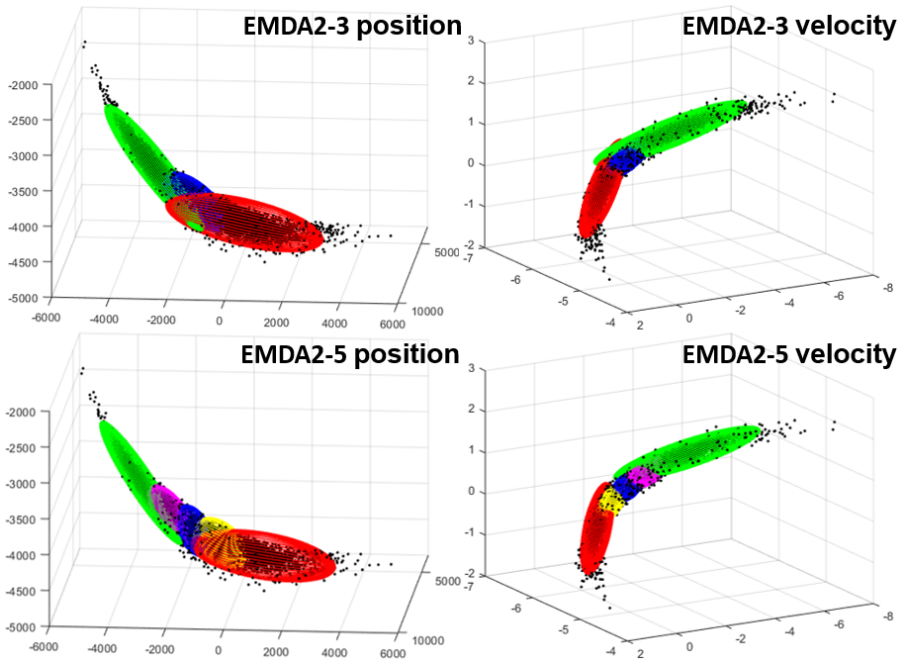

Fig. 8: Position and Velocity EM clustering representation with 3 and 5 Gaussians. First propagation from initial condition in the case of 3 observations each 2 orbits. Portrait of the Gaussian kernels before the measurement update.

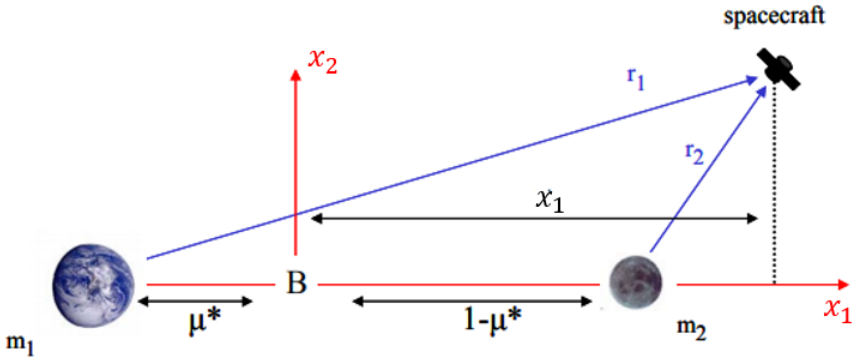

Fig. 9: Coordinates in Earth-Moon system.

such that the distance between the two celestial masses defines the distance unit: 1 DU = 385692,5 km. Under this parametrization, the distances of the

spacecraft with respect to the Earth, r_1, and to the Moon, r_2, are

$$r_1 = \sqrt{(x_1 + \mu^*)^2 + x_2^2 + x_3^2} \tag{45}$$

$$r_2 = \sqrt{(x_1 - 1 + \mu^*)^2 + x_2^2 + x_3^2} \tag{46}$$

The spacecraft, assumed of negligible mass compared to the two bodies, is governed by the following equations of motion [11]:

$$\ddot{x}_1 = 2\dot{x}_2 + x_1 - \frac{(1-\mu^*)(x_1+\mu^*)}{r_1^3} - \frac{\mu^*(x_1 - 1 + \mu^*)}{r_2^3} \tag{47}$$

$$\ddot{x}_2 = 2\dot{x}_1 + x_2 - \frac{x_2(1-\mu^*)}{r_1^3} - \frac{x_2\mu^*}{r_2^3} \tag{48}$$

$$\ddot{x}_3 = -\frac{x_3(1-\mu^*)}{r_1^3} - \frac{x_3\mu^*}{r_2^3} \tag{49}$$

The dynamics are propagated at 1.4 Hz with a Runge-Kutta 7-8 integrator. The system has five different equilibrium points, the so-called liberation or Lagrange points. In the presented application, the filter estimates the orbit parameters of the spacecraft around the unstable L2 location, at coordinates (1.156, 0, 0) DU [5].

The initial state distribution is assumed to be Gaussian $\mathbf{x}_0 \sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0)$, with mean at L2 with zero velocity and given covariance matrix $\mathbf{P}_0 = 10^{-3}\mathbf{I}_{6\times 6}$.

The measurement model assumes the radial position of the spacecraft w.r.t. the Earth and the line of sight direction of the planet, as in the previous

problem,

$$y_1 = r_1 + \eta_1 \tag{50}$$

$$y_2 = arctan2(\frac{x_2}{x_1 + \mu^*}) + \eta_2 \tag{51}$$

$$y_3 = arcsin\left(\frac{x_3}{r_1}\right) + \eta_3 \tag{52}$$

where η_i, with $i = 1, 2, 3$, is the measurement noise, assumed to be Gaussian. The standard deviation of the error resembles the previous application.

The selected initial condition integrates to a bimodal PDF, caused by the unstable nature of the L2 point. Therefore, depending on the true initial state of the spacecraft, which is randomly selected according to the initial Gaussian PDF, the equations of motion define two family of orbits, as shown in Figure 10. The different orbit realizations are reported in the (x_1, x_2) plane, as a

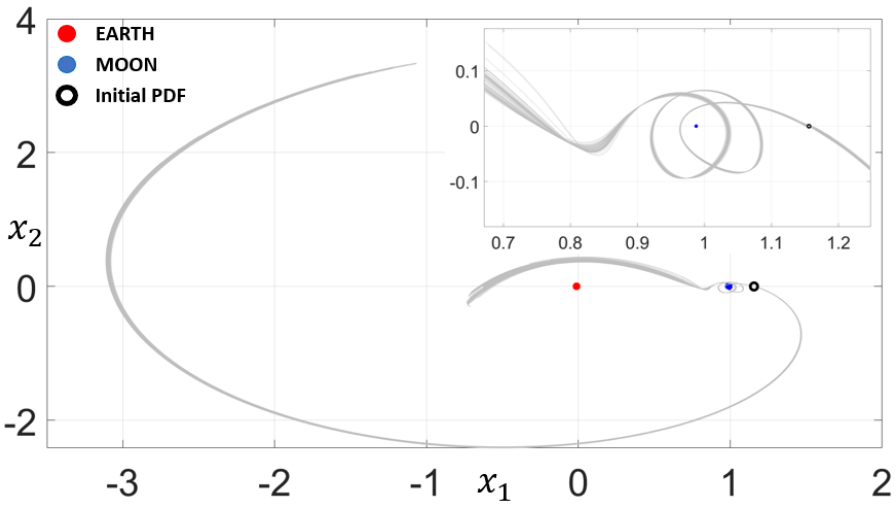

Fig. 10: 200 orbit realizations from L2, propagated for 2.7π.

bidimensional projection of the actual tridimensional orbit. Figure 10 shows that the left part of the initial Gaussian evolves with revolutions around the Moon before leaving due to the gravitational assist. The other half of the initial Gaussian moves away form the Moon and starts orbiting around the Earth.

Process noise is added to the system. The process noise is white and additive to the acceleration of the system. The process noise covariance matrix is computed as [27]

$$\mathcal{T}(t_{k+1}, t_k) = \left[\frac{(t_{k+1} - t_k)^2}{2} \mathbf{I}_3 \quad (t_{k+1} - t_k) \mathbf{I}_3 \right]^T \tag{53}$$

$$\mathbf{Q}_{k+1} = \sigma_Q^2 \mathcal{T}(t_{k+1}, t_k) \mathcal{T}(t_{k+1}, t_k)^T \tag{54}$$

Considering that the state of the system is scaled by the Earth/Moon distance, the value of σ_Q is chosen to be 10^{-6}.

A Monte Carlo analysis with 200 runs has been performed with EMDA2-3 and it is reported in Figure 11. The graphs are zoomed in from the initial large uncertainty levels in order to show convergence. The overlapping, at steady state, between the continuous and dashed blue lines proves the matching between the effective and the estimated uncertainties of the system, expressed as 3σ. Therefore, even in the new scenario, EMDA2-3 is a consistent filter that reaches a precise estimate of the state. The error mean is null as expected by the unbiased nature of the filter.

The performances of EMDA*c*-N is compared against the other filters, as done in the previous problem. Figure 12 reports the standard deviation analy-

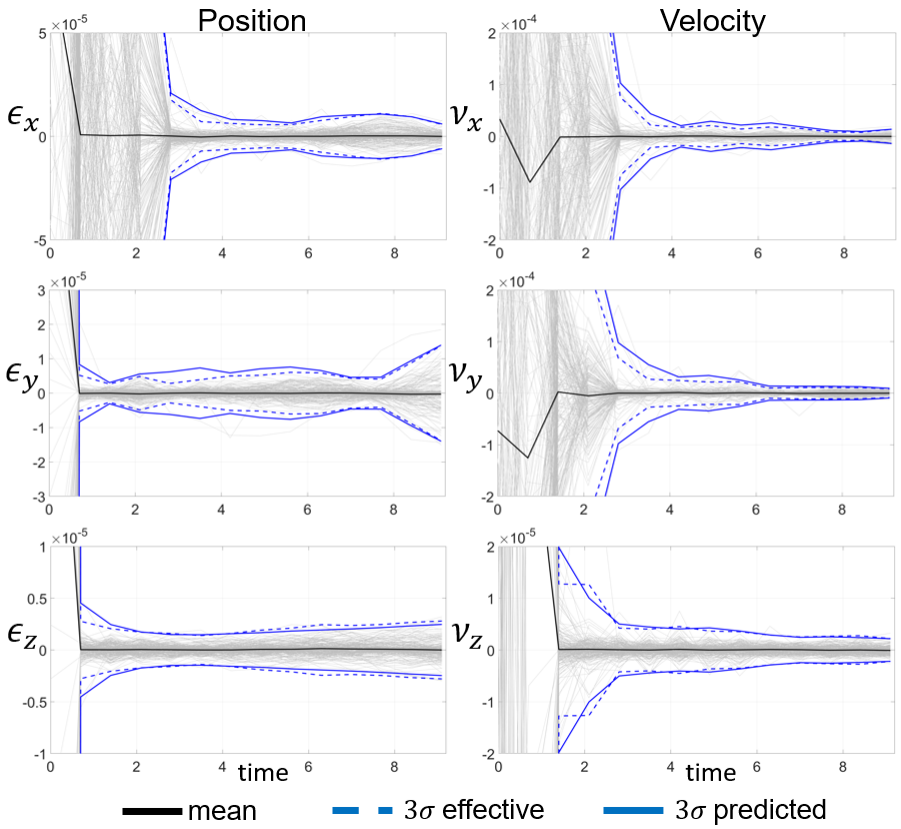

Fig. 11: 400 Monte Carlo runs depicting the performance of EMDA2-3 at 1.4 Hz for 2.7π.

sis for position, left, and velocity, right, of the spacecraft. EMDA2-3 has been compared to the UKF, the BPF, and DAEnKF-2. DAHO-1 and DAEnKF-1 (which reduce to the EKF) diverge; and is therefore not reported. From the figure, it can be noted that the UKF diverges and the effective standard deviation is larger than the filter's own prediction. The DAHO-2 has an analogous behavior to the UKF and it is not reported in the figure. The BPF diverges and the orange curves start growing after the first step. Based on following each single particle, the BPF follows both paths from the initial PDF. Therefore, in

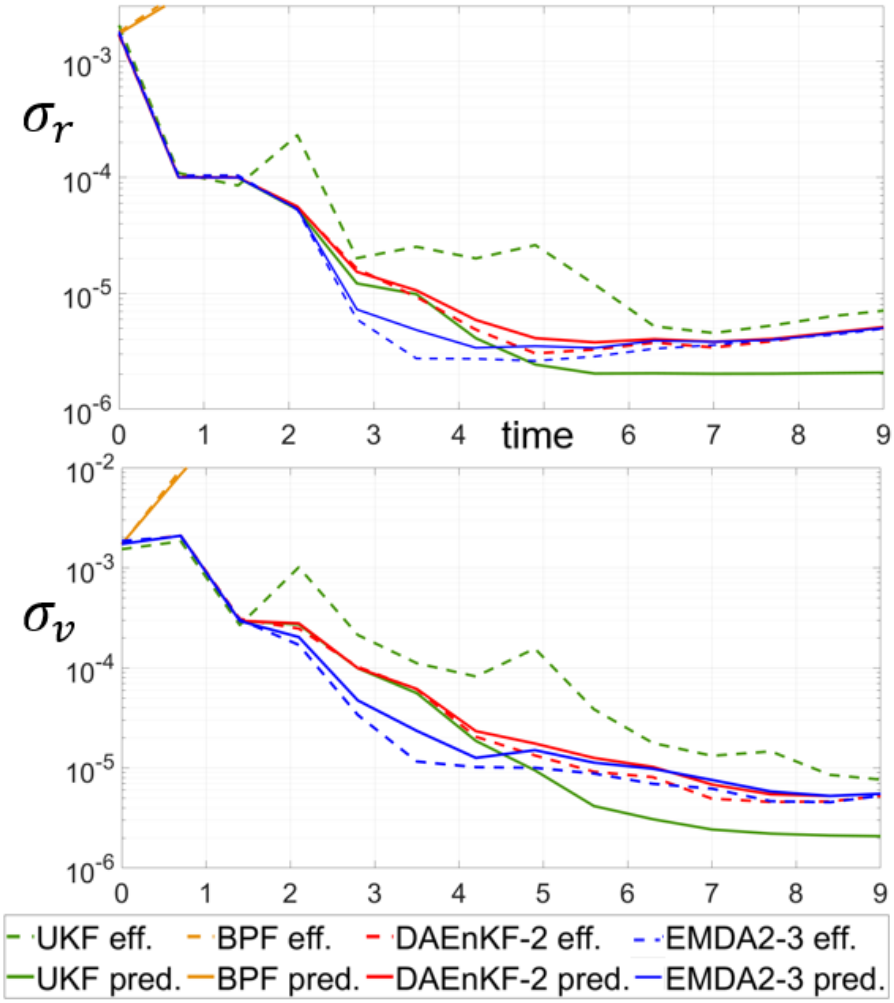

Fig. 12: Position and Velocity error standard deviations comparison with 200 Monte Carlo runs.

the first steps, the corresponding estimate is in conflict on which path of the bifurcation to take, and the filter fails. The remaining two filters, DAEnKF-2 and EMDA2-3, converge and they reach steady state with consistency. These are the filters that approximate the PDFs with clustering, where the measurement likelihood better weights, and picks, which one of the two modes

the true state belongs to. However, as in the previous example, EMDA2-3 shows an improvement in accuracy and its precision levels are superior to its single-Gaussian counterpart. Indeed, the blue lines lie below the red lines during the whole simulation, both for position and velocity. EMDA2-5 achieves convergence with consistency, and it is slightly more accurate than EMDA2-3. However, it is not reported in the figure for clarity purposes.

A computational time analysis is performed on the L2 orbit determination problem in order to underline the benefits of using DA evaluation techniques for the propagation of an ensemble of points. The time analysis studies the average computational time requested by the processor to perform one single run of the Monte Carlo analysis:

$$\tau = \frac{\sum_i^{MC} \sum_j^{N_{steps}} t_{iter}}{MC} \tag{55}$$

where t_{iter} is the computational time requested for one single iteration of the algorithm, N_{steps} is the total number of steps for each Monte Carlo run, and MC is the number of runs. Table 1 reports the values of τ among different filters and for three sizes of particles in the ensemble. The table reports the BPF, EMDA2-3, EMDA2-5, DAEnKF-2, and the MCKF. The MCKF is the classical Monte Carlo Kalman Filter, implemented without differential algebra techniques. Consequently, the particles in MCKF are individually propagated. Figure 13 gives a visual representation of τ through a bar graph. The figure proves that DA reduces the computational time of the filtering algorithm. The three filters based on DA are the fastest, while the BPF is the slowest. DA uses

Filter	$1 \cdot 10^4$	$2 \cdot 10^4$	$3 \cdot 10^4$
BPF	11.798	21.454	35.512
MCKF	12.022	20.135	26.915
EMDA2-5	8.784	17.504	26.698
EMDA2-3	6.613	13.096	19.432
DAEnKF-2	1.659	3.311	4.865

Table 1: Computational time τ analysis among different filters for different ensemble size.

the polynomial map of deviations to propagate all the points with a singe DA integration and n evaluations, while classic particle based filtering techniques, such as the BPF and the MCKF, perform n propagations. Therefore, the main advantage of leveraging DA is appreciated by comparing the DAEnKF-2 computational time with that of the MCKF. The two filters have the same accuracy and robustness levels, but DAEnKF-2 is considerably faster. It is important to re-emphasize that DAEnKF-c is equivalent to EMDAc-1, by selecting $N = 1$ the filter skips the clustering (K-means and the EM) part of its algorithm. Therefore, in the figure, DAEnKF-2 is the fastest algorithm, and the computational time increases as we increase the number of Gaussian kernels. Increasing the number of kernels produces a more precise estimate at the cost of a heavier computational burden. As expected, as the number of particles n becomes larger, the computational effort increases as well. The value of τ requested by EMDAc-N depends also on the tolerance selected to stop the EM iterations. Initializing EM with the output of K-means typically requires only few iterations.

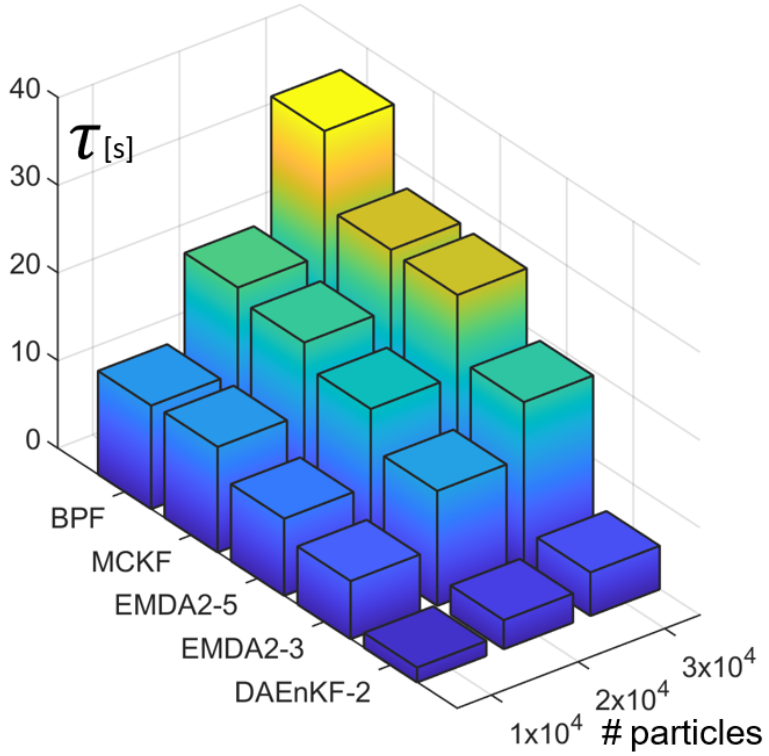

Fig. 13: Computational time comparison among different filters.

6 Conclusions

A new differential algebra particle filter has been presented. The nonlinearity of the dynamics and measurements is approximated by high order Taylor series expansions using differential algebra (DA) techniques. Using a truncated Taylor series representation of the dynamics, the propagation step of each particle is replaced with a faster polynomial evaluation. Working in the DA framework significantly reduces the computational burden required by standard approaches to propagate particles forward in time [7] [19], while retaining high accuracy [24] [20]. The proposed algorithm, named EMDAc-N, utilizes

soft clustering, thought Expectation Maximization. As a consequence, during the measurement update, each model works with the whole set of points. Clustering the propagated PDF improves the estimation of the state of the system: multiple models better represent the shape of the distribution especially for long time intervals, better than a single Gaussian. In the numerical examples considered, it is shown that increasing the number of Gaussian components achieves satisfactory accuracy and robustness levels in the challenging situation of having sporadic measurements, where filters with a Gaussian approximation fail to converge.

Acknowledgment

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

References

1. R. Armellin, P. Di Lizia, F. Bernelli-Zazzera, and M. Berz. Asteroid close encounters characterization using differential algebra: the case of apophis. *Celestial Mechanics and Dynamical Astronomy*, 107(4):451–470, 2010. doi: 10.1007/s10569-010-9283-5.
2. D. Arthur and S. Vassilvitskii. K-means++: the advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.
3. F. Cavenago, P. Di Lizia, M. Massari, S. Servadio, and A. Wittig. Da-based nonlinear filters for spacecraft relative state estimation. In *2018 Space Flight Mechanics Meeting*, page 1964, 2018. doi: 10.2514/6.2018-1964.
4. R. Chen and J. S. Liu. Mixture Kalman filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(3):493–508, 2000.

5. H. D. Curtis. *Orbital mechanics for engineering students*. Butterworth-Heinemann, 2013. pages. 126-133.
6. K. J. DeMars, R. H. Bishop, and M. K. Jah. Entropy-based approach for uncertainty propagation of nonlinear dynamical systems. *Journal of Guidance, Control, and Dynamics*, 36(4):1047–1057, July–August 2013.
7. P. Di Lizia, M. Massari, and F. Cavenago. Assessment of onboard DA state estimation for spacecraft relative navigation. Technical report, European Space Agency, 2017.
8. A. Gelb, editor. *Applied Optimal Estimation*. The MIT press, Cambridge, MA, 1974. ISBN:9780262200271.
9. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, NY, 2001.
10. A. J. Haug. A tutorial on Bayesian estimation and tracking techniques applicable to nonlinear and non-Gaussian processes. Technical report, MITRE Corporation, McLean, Virginia, 2005.
11. K. C. Howell. Three-dimensional, periodic, 'halo' orbits. *Celestial mechanics*, 32(1):53–71, 1984.
12. B. A. Jones and R. Weisman. Multi-fidelity orbit uncertainty propagation. *Acta Astronautica*, 155:406–417, 2019.
13. S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, March 2004. doi: 10.1109/JPROC.2003.823141.
14. S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of 1995 American Control Conference-ACC'95*, volume 3, pages 1628–1632. IEEE, 1995.
15. J. Junkins and P. Singla. How nonlinear is it? A tutorial on nonlinearity of orbit and attitude dynamics. *Journal of the Astronautical Sciences*, 52(1-2):7–60, 2004.
16. R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(Series D):35–45, March 1960. doi:10.1115/1.3662552.
17. R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction. *Journal of Basic Engineering*, 83(Series D):95–108, March 1961. doi:10.1115/1.3658902.
18. B. M. *Modern Map Methods in Particle Beam Physics*. Academic Press, 1999.

19. M. Massari, P. Di Lizia, F. Cavenago, and A. Wittig. Differential algebra software library with automatic code generation for space embedded applications. In *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, 2018. doi: 10.2514/6.2018-0398.

20. M. Massari, P. Di Lizia, and M. Rasotto. Nonlinear uncertainty propagation in astrodynamics using differential algebra and graphics processing units. *Journal of Aerospace Information Systems*, 14(9):493–503, 2017.

21. R. Park and D. Scheeres. Nonlinear mapping of Gaussian statistics: theory and applications to spacecraft trajectory design. *Journal of guidance, Control, and Dynamics*, 29(6):1367–1375, 2006. doi: 10.2514/1.20177.

22. R. S. Park and D. J. Scheeres. Nonlinear semi analytic methods for trajectory estimation. *Journal of Guidance Control and Dynamics*, 30(6):1668–1676, 2007. doi: 10.2514/1.29106.

23. D. Raihan and S. Chakravorty. Particle Gaussian Mixture (PGM) filters. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1369–1376, July 2016.

24. M. Rasotto, A. Morselli, A. Wittig, M. Massari, P. Di Lizia, R. Armellin, C. Valles, and G. Ortega. Differential algebra space toolbox for nonlinear uncertainty propagation in space dynamics. In *The 6th International Conference on Astrodynamics Tools and Techniques (ICATT)*, 03 2016.

25. S. Servadio, R. Zanetti, and B. A. Jones. Nonlinear Filtering with a Polynomial Series of Gaussian Random Variables. *IEEE Transactions on Aerospace and Electronic Systems*, 57(1):647–658, February 2021. DOI: 10.1109/TAES.2020.3028487.

26. A. Smith. *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2013.

27. B. D. Tapley, B. E. Schutz, and G. H. Born. *Statistical Orbit Determination*. Elsevier Academic Press, 2004. ISBN: 9780080541730.

28. G. Terejanu, P. Singla, T. Singh, and P. D. Scott. Adaptive Gaussian Sum Filter for Nonlinear Bayesian Estimation. *IEEE Transactions on Automatic Control*, 56(9):2151–2156, 2011.

29. M. Valli, R. Armellin, P. Di Lizia, and M. Lavagna. A Gaussian particle filter based on differential algebra for spacecraft navigation. In *Proceedings of the International Astronautical Congress, IAC*, volume 7, 10 2012.

30. M. Valli, R. Armellin, P. Di Lizia, and M. Lavagna. Nonlinear mapping of uncertainties in celestial mechanics. *Journal of Guidance, Control, and Dynamics*, 2013. doi: 10.2514/1.58068.

31. M. Valli, R. Armellin, P. Di Lizia, and M. R. Lavagna. Nonlinear filtering methods for spacecraft navigation based on differential algebra. *Acta Astronautica*, 94(1):363–374, 2014. doi: 10.1016/j.actaastro.2013.03.009.

32. A. Wittig, P. Di Lizia, R. Armellin, K. Makino, F. Bernelli-Zazzera, and M. Berz. Propagation of large uncertainty sets in orbital dynamics by automatic domain splitting. *Celestial Mechanics and Dynamical Astronomy*, 122(3):239–261, 2015.

33. S. Yun and R. Zanetti. Nonlinear filtering of light-curve data. *Advances in Space Research*, 66(7):1672–1688, October 2020. doi: 10.1016/j.asr.2020.06.024.