

# DA-BASED MULTIPLE GAUSSIANS PARTICLE FILTER FOR ORBIT DETERMINATION

Simone Servadio\*, Renato Zanetti†

The nonlinear filtering problem plays a fundamental role in multiple space related applications. This paper offers a new filtering technique that combines the classical ensemble Kalman filter with a Gaussian mixture model. 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), as a clustering algorithm, leads to a better approximation of the propagated probability density function and to a multiple weighted measurement update. The performances of the new method is assessed in the nonlinear Orbit Determination problem, for the challenging case of low observations frequency.

## INTRODUCTION

The filtering problem for nonlinear dynamic system is an important research area that has attracted considerable interest, especially in space applications: it consists of estimating the state of a nonlinear dynamical system from noisy measurements. For the well-know linear and Gaussian case, the posterior distribution remains Gaussian and the Kalman Filter [1, 2] provides the mechanization to calculate its mean and covariance. However, most problems of interest in aerospace engineering applications, such as orbit determination [3], spacecraft navigation, target tracking, etc., require a reliable nonlinear filtering method that deals with high nonlinearities. In the presence of nonlinearities, the posterior distribution is necessarily a non-Gaussian probability density function (PDF) that is typically not representable exactly in closed form and needs to be approximated.

The most common solution for the nonlinear estimation problem is the Extended Kalman Filter (EKF) [4]. 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 [5]. In such cases, a different approach that accounts for the system nonlinearities must be used. The Unscented Kalman Filter (UKF) [6, 7] 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 [8, 9] 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. [10] reproduced Park and Sheeres' work using Differential Algebra (DA), eliminating the need to evaluate STT.

Raihan and Chakravorty [11] developed a particle Gaussian mixture filter similar to the one that has been proposed in this paper. Their estimator has the usual computational burden that characterize all particle filter: having to propagate the whole set of samples one by one. Moreover, by using K-means as their clustering

---

\*Ph.D. Candidate, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin

†Assistant Professor, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin

algorithm, the covariance calculation from one single cluster is performed using only the points belonging to that cluster, without any influence from the rest of the ensemble.

Gaussian Sum Filters (GSF) divide the uncertainties domain with multiple Gaussian kernels and apply the EKF equations for each model. The Gaussian Multiple Model (GMM) final estimate is a weighted sum of the estimates from each model, based on the measurement likelihood function. However, the drawback of a linear propagation is still present and can result in convergence issues with sporadic measurements. Accuracy during the PDF time-update has been improved by changing the weights of each distribution regardless the availability of measurements [12], or by changing the number of kernels to better address the nonlinearities present in the system [13].

The filters above are more accurate than the EKF, but the evaluation of STT or performing multiple high fidelity integrations, lead to computational complexity [14]. High accuracy can also be achieved with sequential Monte Carlo methods (the particle filter, PF)[15], or the ensemble Kalman filter (EnKF) [16]. However, even if these filters can provide a good estimate of the states and uncertainties, performing an elevated number of propagations is computationally very expensive and, as a consequence, they do not fit for orbit determination problems or on-board applications.

Valli et al. [17] use DA techniques to develop an ensemble Kalman filter that substitutes samples propagation with polynomial evaluations: thus it enhances and speeds up the classical Monte Carlo approach. However, their update is singular, i.e. it has only one kernel, since the distributions are approximated with one single Gaussian, neglecting 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 [17], but it improves on the update with the introduction of multiple Gaussian models.

This paper introduces a filter,  $EMDA_{n-k}$ , that combines the strengths of the above cited filters. EMDA will use DA techniques to solve the prediction problem through DA based Monte Carlo simulations. 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 (represented using Taylor series expansion up to a user-defined order  $n$ ). 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,  $k$  updates are performed and the estimate is evaluated through weighted mean.

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 filter 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. Furthermore, 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. Lastly, conclusions are deduced.

## DIFFERENTIAL ALGEBRA

Differential Algebra techniques are used to obtain the  $c$ -th order Taylor expansion series of the flow from 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 mere 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. 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.

An implementation of such DA computer routines is available in the Differential Algebra Core Engine

(DACE 2.0) software, which is been used to implement the algorithm presented in this paper. For additional explanation on DA, the reader is advised to look through previous works that include a satisfactory introduction such as [10] and [19].

## 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 and where the only information about the system is a set of measurements, related to the state vector, acquired at discrete times.

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

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

where  $\mathbf{f}$  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  $\mathbf{h}$  is the measurement function. The process noise  $\mathbf{v}$  and the measurement noise  $\mathbf{w}$  are Gaussian zero-mean random sequences which satisfy the conditions  $\forall i, j > 0$ :

$$\mathbb{E} \{ \boldsymbol{\nu}(i) \} = \mathbb{E} \{ \boldsymbol{\eta}(i) \} = \mathbf{0} \quad (3)$$

$$\mathbb{E} \{ \boldsymbol{\nu}(i) \boldsymbol{\nu}^T(j) \} = \mathbf{Q} \delta_{ij} \quad (4)$$

$$\mathbb{E} \{ \boldsymbol{\eta}(i) \boldsymbol{\eta}^T(j) \} = \mathbf{R} \delta_{ij} \quad (5)$$

$$\mathbb{E} \{ \boldsymbol{\nu}(i) \boldsymbol{\eta}^T(j) \} = \mathbf{0} \quad (6)$$

### Cluster Propagation

The truncation order of the Taylor expansion series can be arbitrary picked, it is denoted by  $c$ . Let  $N$  be the fixed number of Gaussian distributions that will approximate the samples disposition of the particle filter. Therefore, each Gaussian PDF will be described by its own mean  $\hat{\mathbf{x}}_i$  and covariance matrix  $\mathbf{P}_i$ , and it will have an importance weight  $\mu_i$ , with  $i = 1, \dots, N$ . The sum of the weights gives unit value and the weighted mean of the Gaussians means is the current estimate. 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 iteration starts with sampling from the GMM, in particular each  $i^{th}$  Gaussian component generates a number of particles proportional to its weight

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

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.

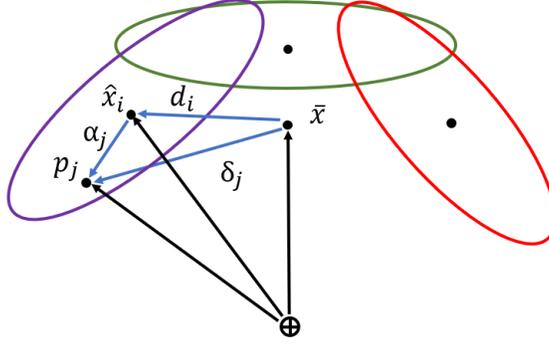
The propagation is carried out in the DA framework. Differential Algebra creates a map of the dynamics that connects deviations at time steps  $k$  to deviations at time step  $k + 1$ . Therefore, it is sufficient to propagate the mean of the samples and, subsequently, evaluate the polynomial map for each deviation:  $n$  propagations have been carried out with only one propagation, the current estimate, and  $n$  polynomial evaluations. The results is a computational efficient way to evaluate the predicted PDF.

The next step is to calculate the deviation of each single sample from its component's mean: this is achieved through simple vector subtraction. Looking at Figure 1, the deviation  $\delta_j$ ,  $j = 1, \dots, n$ , of each sample from the global mean  $\bar{\mathbf{x}}_k$  is the vector summation of its own deviation from the mean of the Gaussian it belongs to,  $\alpha_j$ , and the distance between that Gaussian mean and the estimate,  $d_i$ .

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

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

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



**Figure 1. Vectors representation.**

where  $p_{j,k}$  is the sample 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.

$$\mathbf{x}_k = \bar{\mathbf{x}}_k + \delta \mathbf{x} \quad (11)$$

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

where  $\mathbf{x}_{k+1}^-$  indicates the Taylor expansion series of the dynamics centered in  $\bar{\mathbf{x}}_k$  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 particles of the propagated PDF is found as

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

### K-means

As with all particle filters, the propagated PDF is represented with  $n$  samples. The next step is to approximate the distribution as a GMM [11]. From the GMM is possible to incorporate the measurements with the Gaussian Sum Filter algorithm. Expectation Maximization (EM) is an algorithm that evaluates means and covariances of the Gaussian distributions to approximate the cluster. However, it is desirable to start the algorithm with a good initial guess of the means in order to significantly reduce the number of iterations required and, consequently, the computational burden. The best starting guess for the EM clustering algorithm is the output of K-means clustering. K-means divides the cluster 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 single sample has a probability of belonging to each different set.

Therefore, after calculating the predicted mean  $\bar{\mathbf{x}}_{k+1}^-$  and covariance  $\mathbf{P}_{k+1}^-$  of the propagated cluster, we 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:

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

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

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 output are the  $N$  state vectors  $\hat{\mathbf{x}}_{i,k+1}^-$ .

### Expectation Maximization (EM)

As stated above, in the EM algorithm each single sample has an associated weight to each Gaussian PDF that indicates how much it is likely to belong to that distribution. The weights are then normalized to sum to unity. The initial mean and covariance are selected according to the K-means algorithm.

Therefore, assuming Gaussian distributions, each particle has a probability to belong to each of the  $N$  Gaussians given by

$$\begin{aligned} P(p_{j,k+1}|i) &= \mathcal{N}\left(\hat{\mathbf{x}}_{i,k+1}^-, \mathbf{P}_{i,k+1}^-\right) = \\ &= \frac{1}{(2\pi)^{s/2} \sqrt{\det \mathbf{P}_{i,k+1}^-}} \exp\left(-\frac{1}{2} \left(p_{j,k+1} - \hat{\mathbf{x}}_{i,k+1}^-\right) \left(\mathbf{P}_{i,k+1}^-\right)^{-1} \left(p_{j,k+1} - \hat{\mathbf{x}}_{i,k+1}^-\right)^T\right) \end{aligned} \quad (16)$$

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

$$P(i|p_{j,k+1}) = \frac{P(p_{j,k+1}|i)P(i)}{\sum_{h=1}^N P(p_{j,k+1}|h)P(h)} \quad (17)$$

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 (16), 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.

$$\begin{aligned} w_{i,j} &= \frac{P(p_{j,k+1}|i)}{\sum_{h=1}^N P(p_{j,k+1}|h)} \\ &= \frac{1}{\sum_{h=1}^N \exp(\log[P(p_{j,k+1}|h)] - \log[P(p_{j,k+1}|i)])} \quad \forall i = 1, \dots, N \wedge \forall j = 1, \dots, n \end{aligned} \quad (18)$$

Conceptually,  $w_{i,j}$  defines how much the  $j$ -th sample belongs to the  $i$ -th 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.

$$\mathcal{W}_i = \sum_{j=1}^n w_{i,j} \quad (19)$$

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

$$\hat{\mathbf{x}}_{i,k+1}^- = \frac{1}{\mathcal{W}_i} \sum_{j=1}^n w_{i,j} p_{j,k+1} \quad (20)$$

$$\mathbf{P}_{i,k+1}^- = \frac{1}{\mathcal{W}_i} \sum_{j=1}^n w_{i,j} \left(p_{j,k+1} - \hat{\mathbf{x}}_{i,k+1}^-\right) \left(p_{j,k+1} - \hat{\mathbf{x}}_{i,k+1}^-\right)^T \quad (21)$$

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

$$\mu_{i,k+1}^- = \frac{\mathcal{W}_i}{\sum_{h=1}^N \mathcal{W}_h} \quad (22)$$

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.

## Measurement Update

The prediction part of the filter is over as the propagated PDF of the state has been approximated as a GMM. The measurement update is now performed via  $N$  updates for the components followed by the update on the weights.

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

$$\mathbf{z}_{k+1} = \mathbf{h}[\mathbf{x}_{k+1}] \quad (23)$$

Where  $\mathbf{z}_{k+1}$  is, again, a polynomial centered in  $\bar{\mathbf{x}}_k$  truncated at order  $c$ . The polynomial can now be evaluated using the deviations for Equation (10) in order to calculate the predicted measurements associated with each particle. Therefore, not only the propagation, but also the measurement evaluation is computationally fast when working in the DA framework.

$$q_{j,k+1} = \mathbf{z}_{k+1}(\delta_j) \quad (24)$$

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:

$$\hat{\mathbf{z}}_{i,k+1}^- = \frac{1}{\mathcal{W}_i} \sum_{j=1}^n w_{i,j} q_{j,k+1} \quad (25)$$

$$\mathbf{P}_{ZZ,i} = \frac{1}{\mathcal{W}_i} \sum_{j=1}^n w_{i,j} \left( q_{j,k+1} - \hat{\mathbf{z}}_{i,k+1}^- \right) \left( q_{j,k+1} - \hat{\mathbf{z}}_{i,k+1}^- \right)^T + \mathbf{R} \quad (26)$$

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

It is important to underline that the covariances of each model uses the whole ensemble of points, and then scale the results according to the model importance. The PGM filter developed in [11] uses K-means (hard clustering) and therefore only the points belonging to each cluster are utilized to calculate the component's mean and covariances. This difference arises from adopting soft constraints, instead of hard ones, in the clustering algorithm. As a consequence, different Gaussians component are updated separately without the influence of the actual shape of the posterior distribution.

The Kalman gains are evaluated.

$$\mathbf{K}_i = \mathbf{P}_{XZ,i} (\mathbf{P}_{ZZ,i})^{-1} \quad (28)$$

Each mean and covariance are updated likewise they are working independently

$$\hat{\mathbf{x}}_{i,k+1}^+ = \hat{\mathbf{x}}_{i,k+1}^- + \mathbf{K}_i \left( \mathbf{y}_{k+1} - \hat{\mathbf{z}}_{i,k+1}^- \right) \quad (29)$$

$$\mathbf{P}_{i,k+1}^+ = \mathbf{P}_{i,k+1}^- - \mathbf{K}_i \mathbf{P}_{ZZ,i} \mathbf{K}_i^T \quad (30)$$

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

## Weights Update and Estimate

The influence of each Gaussian to the updated PDF needs to be updated as well. This step can be achieved considering again Baye's rule and obtaining the posterior distribution of the probability of each Gaussian given the measurements.

$$\begin{aligned} \mu_{i,k+1}^+ &= P(i|\mathbf{Y}_{k+1}) = P(i|\mathbf{y}_{k+1}, \mathbf{Y}_k) = \\ &= \frac{P(i, \mathbf{y}_{k+1}|\mathbf{Y}_k)}{P(\mathbf{y}_{k+1}|\mathbf{Y}_k)} = \frac{P(i, \mathbf{y}_{k+1}|\mathbf{Y}_k)}{\sum_{h=1}^N P(h, \mathbf{y}_{k+1}|\mathbf{Y}_k)} = \\ &= \frac{P(\mathbf{y}_{k+1}|i, \mathbf{Y}_k)P(i|\mathbf{Y}_k)}{\sum_{h=1}^N P(h, \mathbf{y}_{k+1}|\mathbf{Y}_k)} \end{aligned} \quad (31)$$

where  $\mathbf{Y}_k$  indicates all the measurements up to time step  $k$ . Looking at Equation (31), 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  $i$ -th Gaussian.

$$P(\mathbf{y}_{k+1}|i, \mathbf{Y}_k) = \frac{1}{(2\pi)^{m/2} \sqrt{\det \mathbf{P}_{ZZ,i}}} \exp\left(-\frac{1}{2} \left(\mathbf{y}_{k+1} - \hat{\mathbf{z}}_{i,k+1}^-\right) \left(\mathbf{P}_{ZZ,i}\right)^{-1} \left(\mathbf{y}_{k+1} - \hat{\mathbf{z}}_{i,k+1}^-\right)^T\right) \quad (32)$$

Therefore, the weights update equation can be written as

$$\mu_{i,k+1}^+ = \frac{P(\mathbf{y}_{k+1}|i, \mathbf{Y}_k) \mu_{i,k+1}^-}{\sum_{h=1}^N \mu_{h,k+1}^- P(\mathbf{y}_{k+1}|h, \mathbf{Y}_k)} \quad (33)$$

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.

$$\bar{\mathbf{x}}_{k+1} = \sum_{i=1}^N \mu_{i,k+1}^+ \hat{\mathbf{x}}_{i,k+1}^+ \quad (34)$$

$$\mathbf{P}_{k+1} = -\bar{\mathbf{x}}_{k+1} \bar{\mathbf{x}}_{k+1}^T + \sum_{i=1}^N \mu_{i,k+1}^+ \left( \mathbf{P}_{i,k+1}^+ + \hat{\mathbf{x}}_{i,k+1}^+ \hat{\mathbf{x}}_{i,k+1}^{+T} \right) \quad (35)$$

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  $\mathbf{P}_{i,k+1}^+$ .

## ORBIT DETERMINATION

The performance of the filter has been assessed with a nonlinear 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  $\mu$  is the Earth gravitational parameter.

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3} \mathbf{r} \quad (36)$$

The initial condition and uncertainties values resemble the one in [10, 20]. Therefore, each length unit has been scaled by the orbit semi-major axis,  $a = 8788 \text{ km}$ , and by the time  $\sqrt{\frac{a^3}{\mu}}$ .

$$\mathbf{x}_0 = \begin{pmatrix} \mathbf{r}_0 \\ \mathbf{v}_0 \end{pmatrix} = \begin{pmatrix} -0.68787 \\ -0.39713 \\ 0.28448 \\ -0.51331 \\ 0.98266 \\ 0.37611 \end{pmatrix} \quad (37)$$

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:

$$y_1 = r + \eta_1 \quad (38)$$

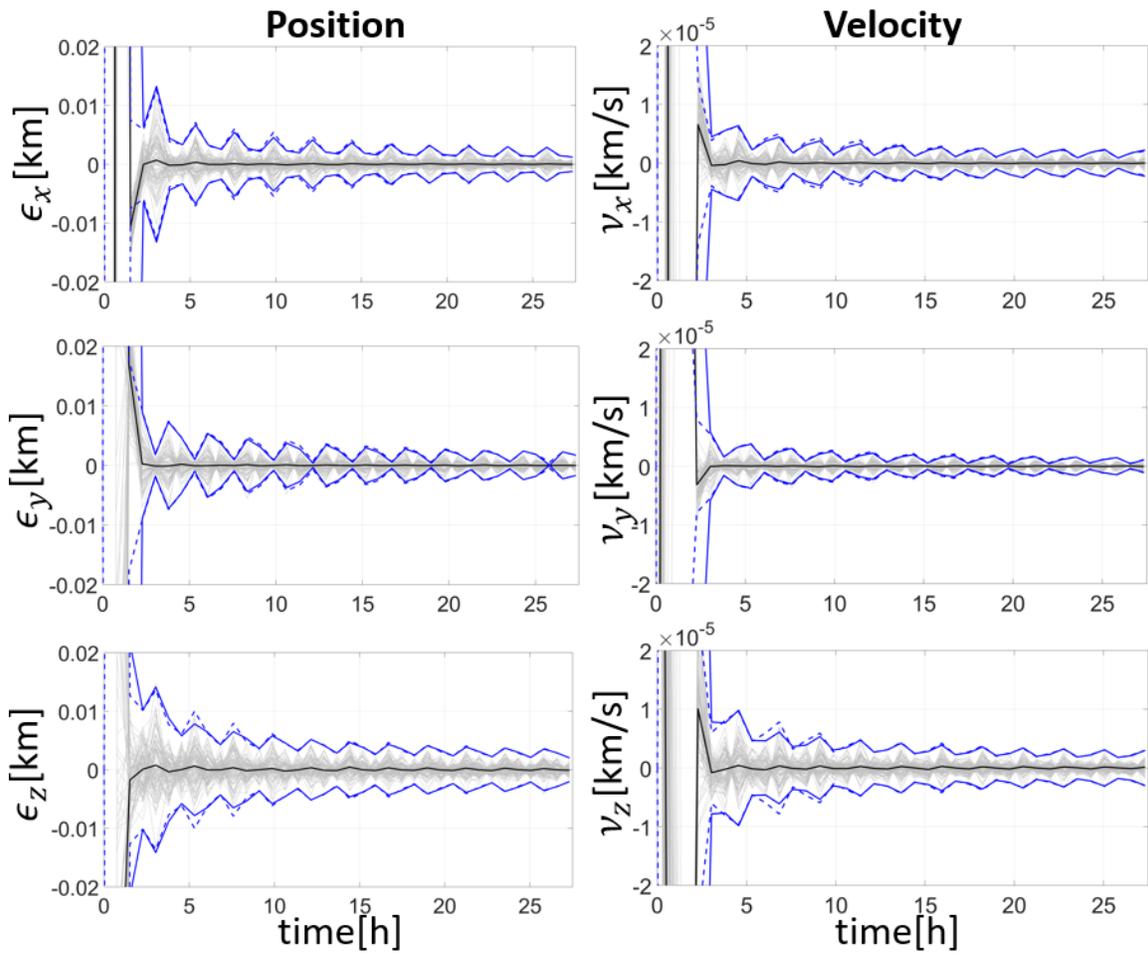
$$y_2 = \arctan2\left(\frac{x_2}{x_1}\right) + \eta_2 \quad (39)$$

$$y_3 = \arcsin\left(\frac{x_3}{r}\right) + \eta_3 \quad (40)$$

where  $\eta_i$ , 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  $\sigma_v = 10^{-4}\sqrt{\frac{\mu}{a}}$ .

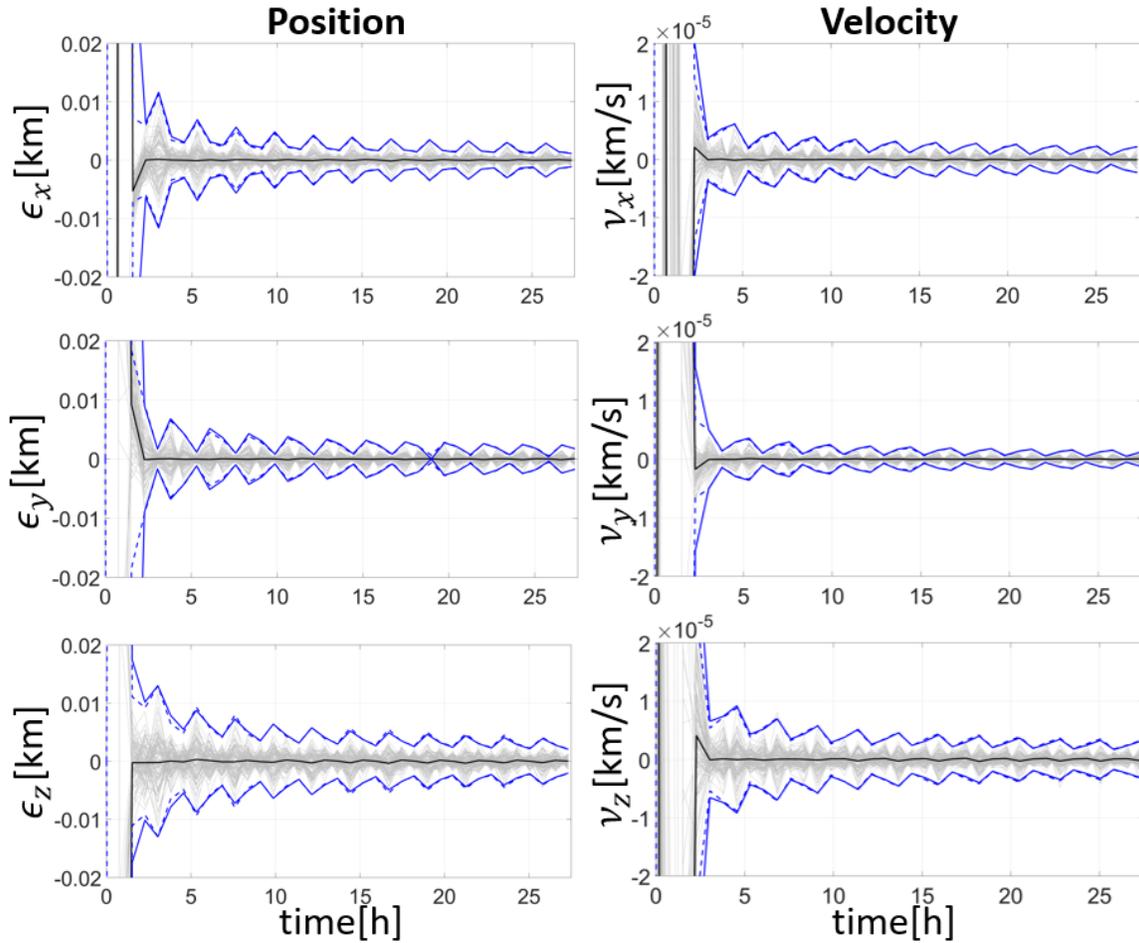
The number of Gaussians in the multiple models is chosen beforehand and it is kept fixed during the whole simulation. Therefore, the name of the filter, EMDA $n$ - $k$  is followed by two different numbers: the first one,  $n$ , indicates the truncation order of the dynamics flow during propagation, while the second one,  $k$ , indicates the number of Gaussians used in the clustering algorithm. According to this notation, EMDA2-3 indicates a filter with truncation order 2 and a GMM update with 3 Gaussians. In each single simulation a total of 10,000 particles have been used.

A Monte Carlo analysis has been performed to assess the consistency of the presented filter and to show reliability and accuracy levels. The simulation has been performed with low acquisition measurement frequency since the clustering algorithm will show its benefits with respect to previous works when the propagated PDF is highly non Gaussian. Thus, Figure 2 and Figure 3 show the performance of EMDA2-3 and EMDA2-5,



**Figure 2. Monte Carlo 100 runs. 12 orbits with 3 observations per orbit. EMDA2-3**

respectively, with a Monte Carlo analysis using 100 runs. The figures show the robustness analysis for 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 con-

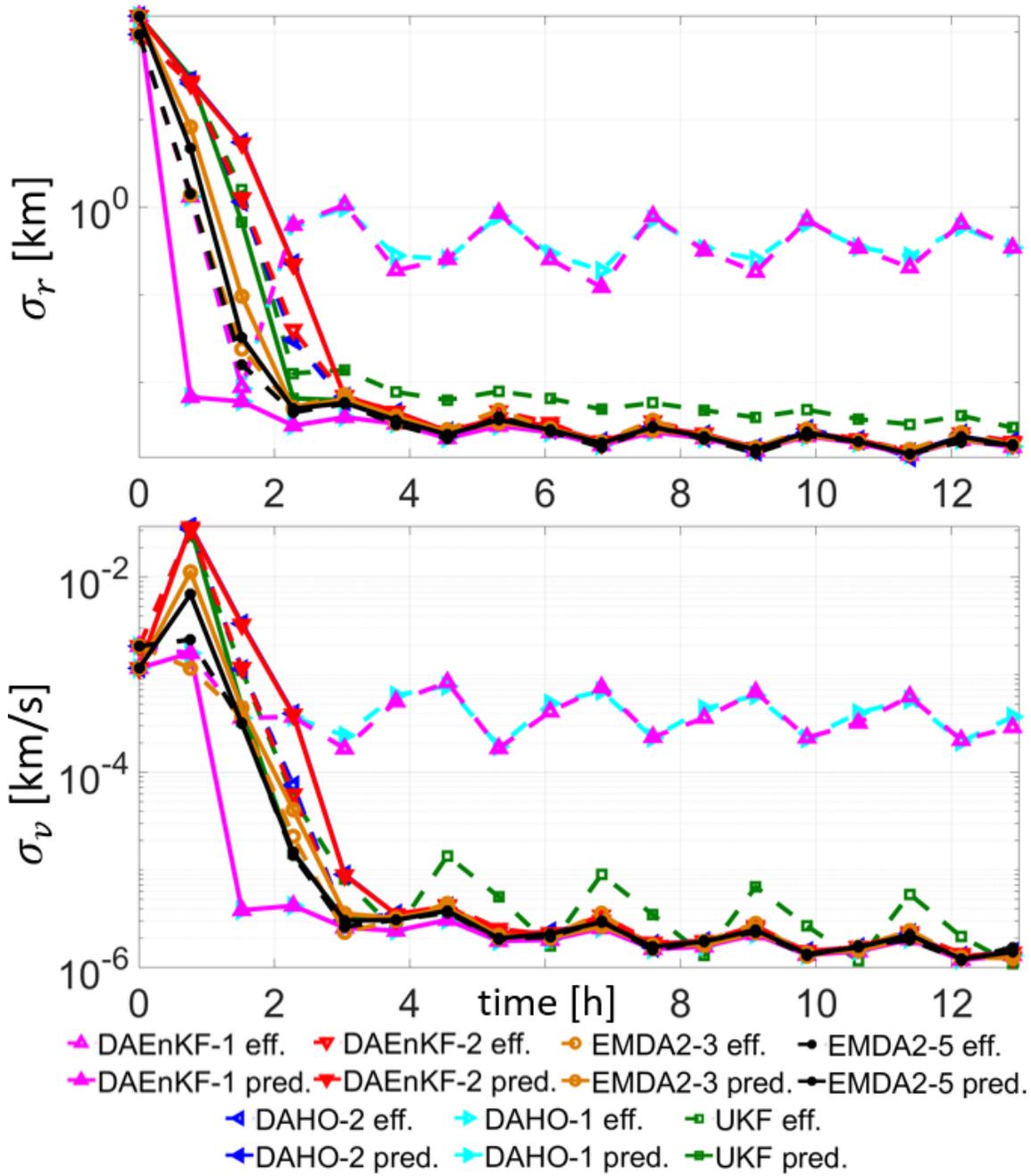


**Figure 3. Monte Carlo 100 runs. 12 orbits with 3 observations per orbit. EMDA2-5**

verge and correctly estimate the state of the system and its uncertainties. Indeed, the continuous blue lines indicate the predicted standard deviations, expressed as  $3\sigma$ , estimated by the filter, while the dashed blue lines represent the actual standard deviations, again as  $3\sigma$ , calculated directly from the Monte Carlo samples, for each time step. 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 null, making EMDA $n-k$  and unbiased filter, in accordance with the theory.

In order to assess the filter accuracy level and robustness, it has been decided to compare it with the most common estimators, such as the EKF and the UKF, and filters from previous works on differential algebra, such as DAHO- $k$  from [10] and DAEnKF from [17]. DAHO- $k$  is a DA-based filter that uses Taylor expansion series up to order  $k$  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, instead, is an Ensemble Kalman Filter that uses polynomial evaluations, through DA, to propagate an ensemble of particles to the next time step, likewise EMDA $n-k$ . Indeed, once the truncation order  $n$  of the Taylor series is selected, DAEnKF corresponds to EMDA $n-1$  where the update has only one model, going back to the classical Ensemble Kalman Filter.

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 stds calculated from the Monte Carlo samples (100 runs), at each



**Figure 4. Position and Velocity error standard deviations comparison. 6 orbits with 3 observations per orbit.**

time step, while the continuous lines are the predicted stds estimated by each filter. These latter are derived from the diagonal terms of the updated covariance matrix at the end of each time-step.

$$\sigma_r = \sqrt{\sigma_{rx}^2 + \sigma_{ry}^2 + \sigma_{rz}^2} \quad (41)$$

$$\sigma_v = \sqrt{\sigma_{vx}^2 + \sigma_{vy}^2 + \sigma_{vz}^2} \quad (42)$$

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

The figure compares, with a logarithmic scale, the estimators mentioned above with different setting parameters, for a total of 7 different filters. Looking at 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. 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 bigger. Looking at the figure, the dashed green line is the only one that does not overlap with the others, both in position and velocity. Nevertheless, the UKF already shows consistency problems with 3 observations per orbit. On the other hand, 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 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- $k$ , 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 furthermore the measurements acquisition frequency in the simulations. Let's now halve the observation to 1.5 per orbit. Figure 5 and Figure 6 show the Monte Carlo analysis, respectively, 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 side, EMDA2-5 has a consistent behavior and both the state and the uncertainties are evaluated correctly, achieving steady state and reaching good accuracy level. Therefore, EMDA $n-k$ , can be compared with the other nonlinear estimators, likewise previously.

Figure 7 represents the standard deviations analysis for DAHO-1, DAHO-2, DAEnKF-1, DAEnKF-2, EMDA2-3, EMDA2-5 and the UKF in a 12 orbits simulation with 1.5 equally spaced observations per orbit. The figure resemble the same characteristics from the 3 observations 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. On the contrary, the continuous lines represent the error standard deviations predicted by the filters according to Equation (41) and (42). The EKF, represented through DAHO-1 and DAEnKF-1, shows the same problems from Figure 4: it diverges and the effective uncertainties increase. 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 $n-k$  is able to estimate the state of the system. When working with multiple Gaussians ( $k > 1$ ), 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.

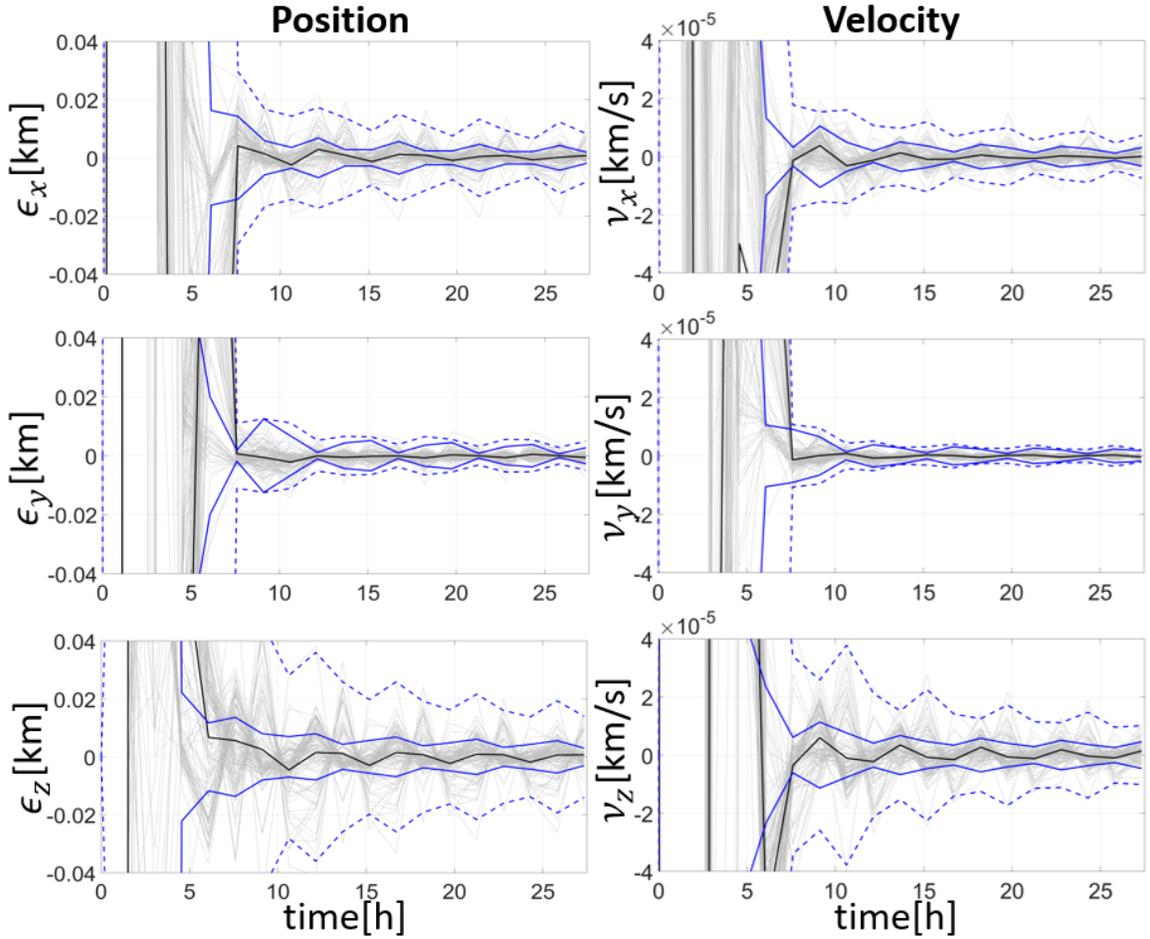


Figure 5. Monte Carlo 100 runs. 12 orbits with 1.5 observations per orbit. EMDA2-3

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 low acquisition frequencies. The clustering algorithm with 5 Gaussians better approximates the shape of the distribution, especially near the mean and at the tails. On the other side, 3 Gaussians struggle to match the curve of the density function and the two normal distribution at the sides overshoot the actual distribution. Indeed, EMDA2-5 achieves a correct estimate when EMDA2-3 fails.

## CONCLUSIONS

A new 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. Thanks to the polynomial representation of the dynamics, the propagation step of each particle has been replaced with a faster polynomial evaluation. Working in the DA framework significantly reduces the computational burden requested by the standard approach [20, 21]. The proposed algorithm, EMDA $n-k$  utilizes soft clustering, thought EM, instead of hard clustering with K-means algorithm as done before. As a consequence, in the 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

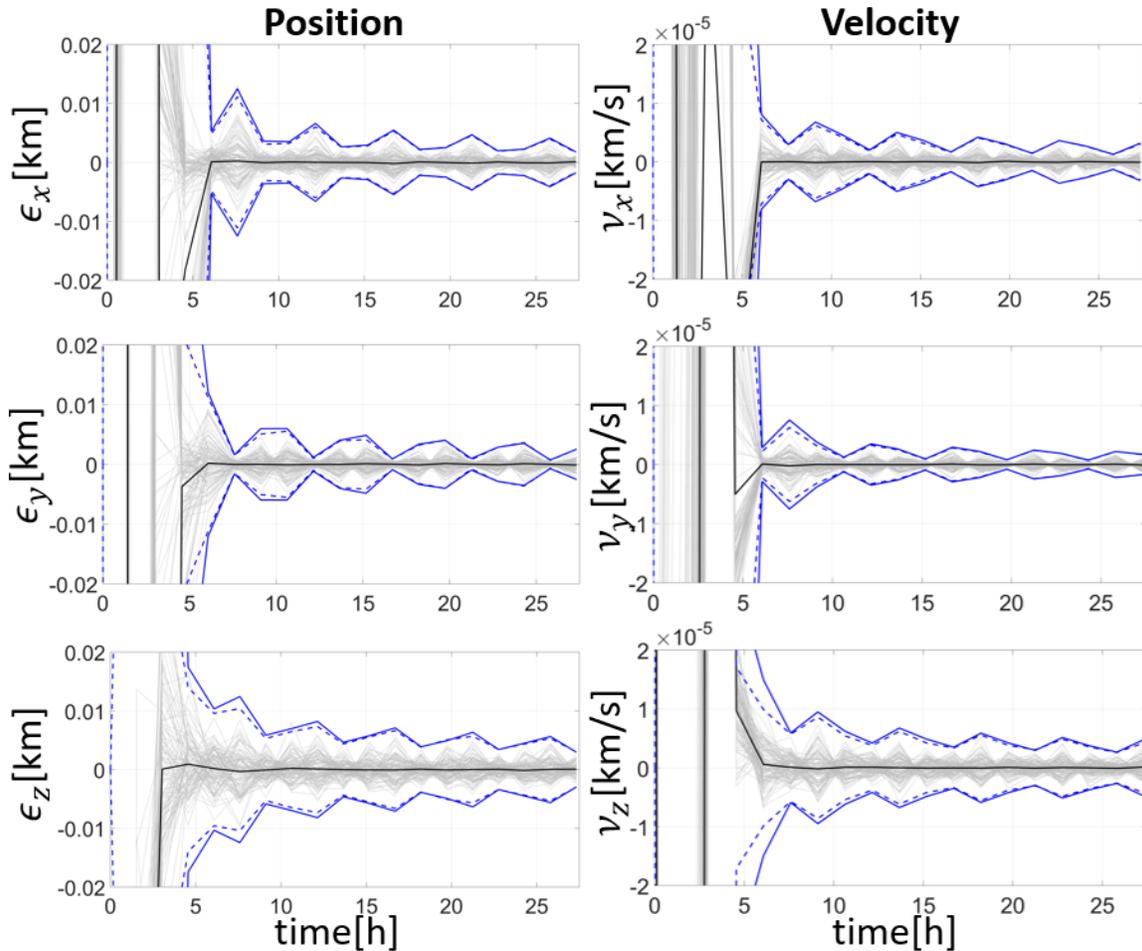
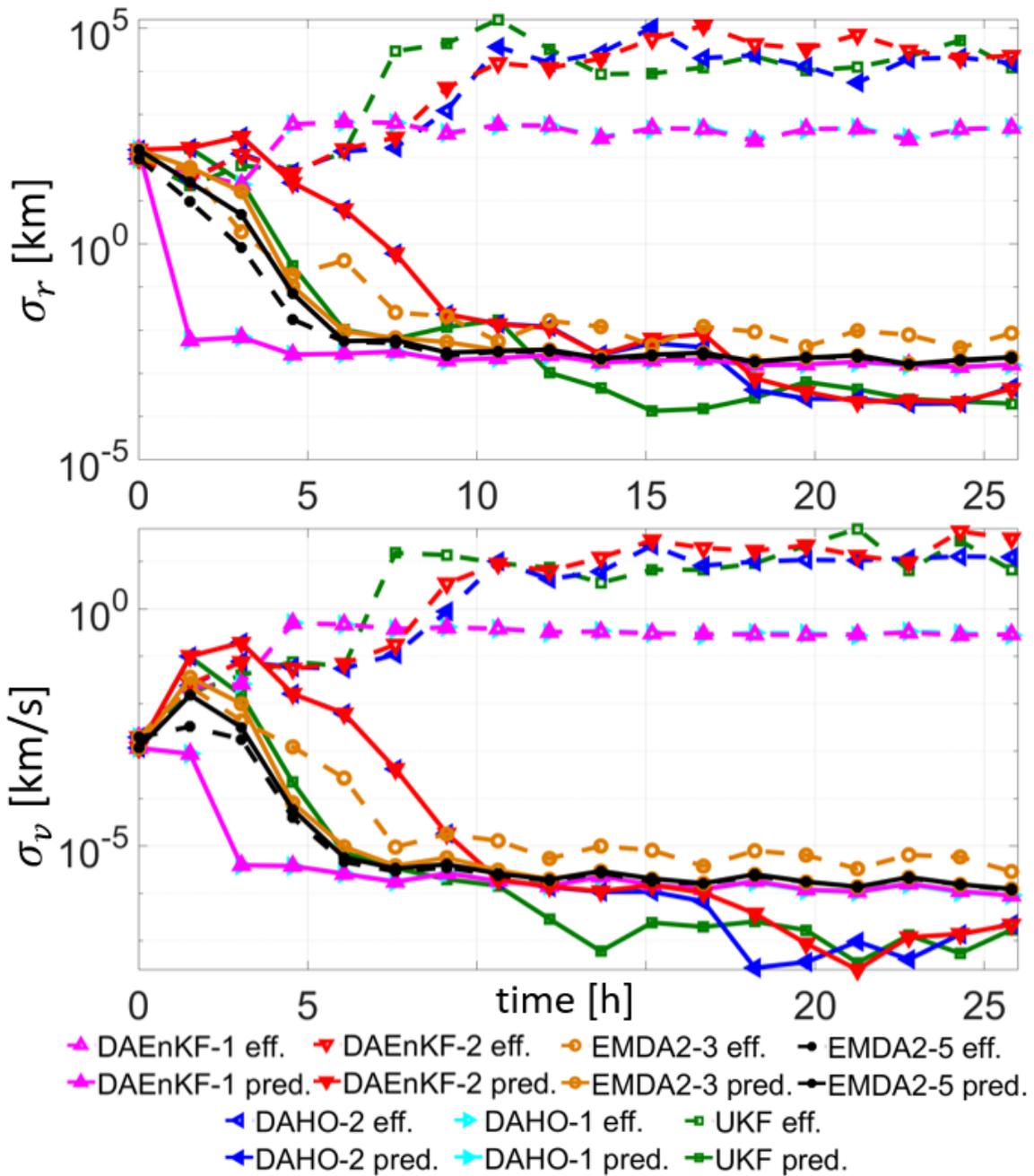


Figure 6. Monte Carlo 100 runs. 12 orbits with 1.5 observations per orbit. EMDA2-5

the challenging situation of having sporadic measurements, where filters with a Gaussian approximation fail to converge. EMDA $n$ - $k$  offers the possibility to achieve convergence by tuning its parameters:  $n$  for the prediction step, and  $k$  for the update.

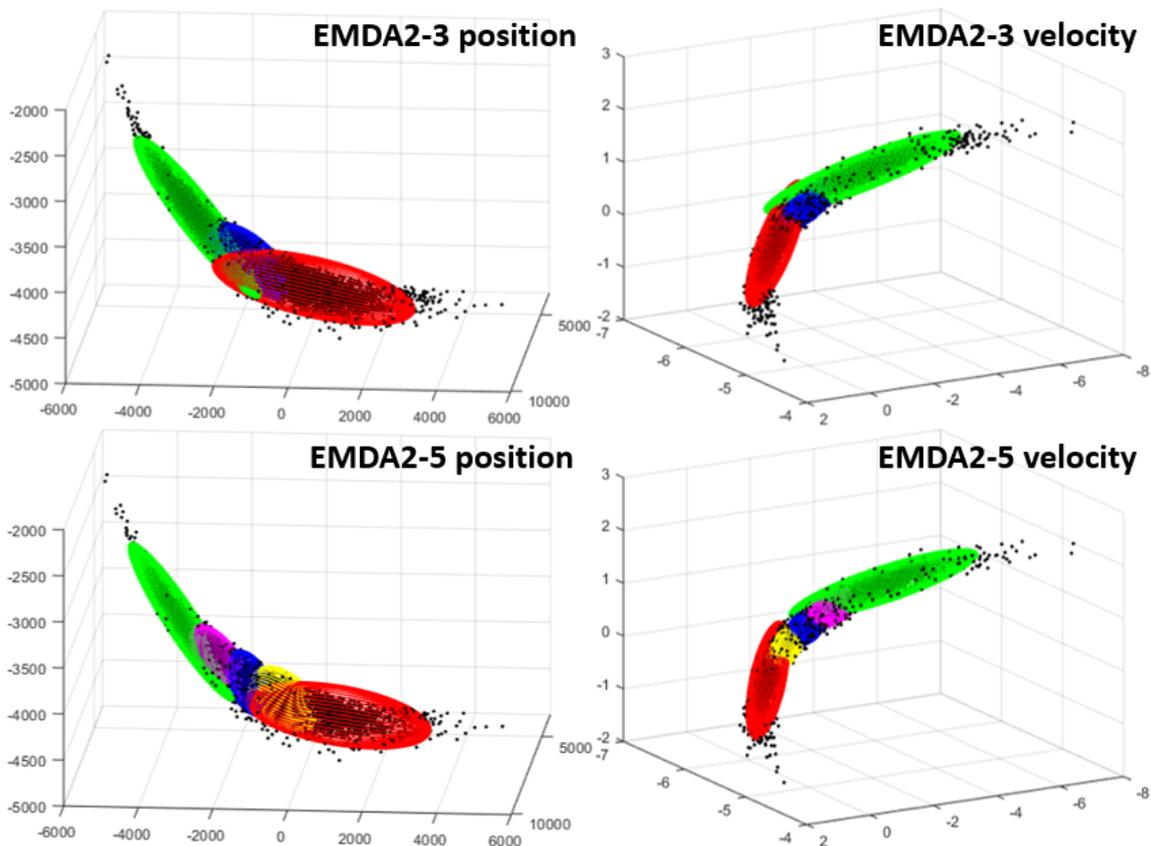
## REFERENCES

- [1] R. E. Kalman, A new approach to linear filtering and prediction problems, *Journal of Basic Engineering* 82 (Series D) (1960) 35–45, doi:10.1115/1.3662552.
- [2] R. E. Kalman, R. S. Bucy, New results in linear filtering and prediction, *Journal of Basic Engineering* 83 (Series D) (1961) 95–108, doi:10.1115/1.3658902.
- [3] B. D. Tapley, B. E. Schutz, G. H. Born, *Statistical Orbit Determination*, Elsevier Academic Press, 2004, ISBN: 9780080541730.
- [4] A. Gelb (Ed.), *Applied Optimal Estimation*, The MIT press, Cambridge, MA, 1974, ISBN:9780262200271.
- [5] J. Junkins, P. Singla, How nonlinear is it? a tutorial on nonlinearity of orbit and attitude dynamics, *Journal of the Astronautical Sciences* 52 (1-2) (2004) 7–60.
- [6] S. J. Julier, J. K. Uhlmann, Unscented filtering and nonlinear estimation, *Proceedings of the IEEE* 92 (3) (2004) 401–422, doi: 10.1109/JPROC.2003.823141.
- [7] S. J. Julier, J. K. Uhlmann, H. F. Durrant-Whyte, A new approach for filtering nonlinear systems, in: *Proceedings of 1995 American Control Conference-ACC'95*, Vol. 3, IEEE, 1995, pp. 1628–1632.



**Figure 7. Position and Velocity error standard deviations comparison. 12 orbits with 1.5 observations per orbit.**

- [8] R. S. Park, D. J. Scheeres, Nonlinear mapping of gaussian statistics: theory and applications to spacecraft trajectory design, *Journal of Guidance, Control, and Dynamics* 29 (6) (2006) 1367–1375, doi: 10.2514/1.20177.
- [9] R. S. Park, D. J. Scheeres, Nonlinear semi analytic methods for trajectory estimation, *Journal of Guidance, Control, and Dynamics* 30 (6) (2007) 1668–1676, doi: 10.2514/1.29106.
- [10] M. Valli, R. Armellin, P. Di Lizia, M. Lavagna, Nonlinear mapping of uncertainties in celestial mechanics, *Journal of Guidance, Control, and Dynamics* 36 (1) (2012) 48–63, doi: 10.2514/1.58068.



**Figure 8. Position and Velocity EM clustering with 3 and 5 Gaussians. First propagation from initial condition; 1.5 observations per orbit.**

- [11] D. Raihan, S. Chakravorty, Particle gaussian mixture (pgm) filters, in: 2016 19th International Conference on Information Fusion (FUSION), 2016, pp. 1369–1376.
- [12] G. Terejanu, P. Singla, T. Singh, P. D. Scott, Adaptive Gaussian Sum Filter for Nonlinear Bayesian Estimation, *IEEE Transactions on Automatic Control* 56 (9) (2011) 2151–2156.
- [13] K. J. DeMars, R. H. Bishop, M. K. Jah, An entropy-based approach for uncertainty propagation of nonlinear dynamical systems, *Journal of Guidance, Control, and Dynamics* 36 (4) (2013) 1047–1057.
- [14] B. A. Jones, R. Weisman, Multi-fidelity orbit uncertainty propagation, *Acta Astronautica* 155 (2019) 406–417.
- [15] A. Smith, *Sequential Monte Carlo methods in practice*, Springer Science & Business Media, 2013.
- [16] P. L. Houtekamer, H. L. Mitchell, Data assimilation using an ensemble kalman filter technique, *Monthly Weather Review* 126 (3) (1998) 796–811.
- [17] M. Valli, R. Armellin, P. Di Lizia, M. Lavagna, A gaussian particle filter based on differential algebra for spacecraft navigation (2012).
- [18] B. M., *Modern Map Methods in Particle Beam Physics*, Academic Press, 1999.
- [19] F. Cavenago, P. Di Lizia, M. Massari, S. Servadio, A. Wittig, Da-based nonlinear filters for spacecraft relative state estimation, in: 2018 Space Flight Mechanics Meeting, 2018, p. 1964, doi: 10.2514/6.2018-1964.
- [20] P. Di Lizia, M. Massari, F. Cavenago, Assessment of onboard da state estimation for spacecraft relative navigation Final report, ESA (2017).
- [21] M. Massari, P. Di Lizia, F. Cavenago, A. Wittig, Differential algebra software library with automatic code generation for space embedded applications, in: 2018 AIAA Information Systems-AIAA Infotech@ Aerospace, 2018, p. 0398, doi: 10.2514/6.2018-0398.