AAS 22-634

DYNAMICS-BASED UNCERTAINTY PROPAGATION WITH LOW-THRUST

Michele Maestrini [*], Andrea De Vittori[†], Pierluigi Di Lizia[‡] and Camilla Colombo[§]

In recent years low-thrust propulsion has become a credible alternative to standard chemical propulsion for many different applications, ranging from orbital transfers to maintenance of small satellites in LEO. Despite the extensive effort that has been put into development of guidance algorithms for low-thrust systems, only few works have considered the implications that such systems may have on Space Situational Awareness. In fact, the stability of thruster performance must be factored-in when computing state uncertainties. These quantities are paramount to characterize conjunction events and ultimately impact operations in terms of acceptable risks and timeliness of collision avoidance maneuver execution. In light of these considerations it becomes a pressing issue to accurately quantify state uncertainties under the effect of low-thrust noise in a timely manner. Hence, it is paramount to select an uncertainty propagation strategy that can deal with high-frequency thrust variations, thrust interruptions, and uncertain command execution times. In fact, these effects, coupled with the nonlinear dynamics and the orbital uncertainties mapping, sway the validity of the standard Gaussianity assumption which is commonly adopted. To face this challenge, this work proposes two separate methods which combine accurate nonlinear techniques with their computationally light linearized counterparts. These approaches have been developed in the framework of the ESA funded ELECTROCAM contract, awarded to the consortium formed by GMV, Politecnico di Milano, and Universidad Carlos III de Madrid. The first technique is called Adaptive Differential Algebraic Gaussian Mixture Model (A-DAGMMo), and it constitutes a DA-based propagation scheme that automatically determines the number of elements necessary to capture the behaviour of the uncertainty during propagation. The second proposed method is the Stochastic Taylor Model (STyMo): an extension of Differential Algebra that provides an expansion of the covariance inflation term linked to the process noise. The results of the application of the presented methods to the uncertainty propagation of several test cases on different orbital regimes is presented in this work.

INTRODUCTION

In recent years low-thrust propulsion has become a credible alternative to standard chemical propulsion for many different applications, ranging from orbital transfers to day-to-day operations of small satellites in LEO.[1] Indeed, this latter case is gaining a growing interest owed to the advent of mega-constellations endowed with a miniaturized propulsion system capable of granting control authority to such a class of satellites.[2] Despite the extensive effort that has been put into development of guidance algorithms relying on low-thrust propulsion, only few works have focused on the analysis of collision avoidance for such missions.[3,4] This task is not only impacted by the standard uncertainty in the orbital dynamics model, but also by the stability of thruster performance. Actually, this aspect must be factored-in when computing state uncertainties in order to adequately estimate the collision probability. Besides, the adoption of low-thrust also affects the time needed for corrective actions due to the reduced control authority that leads to longer thrusting arcs. This in turn impacts the achievable risk reduction or the selection of a suitable threshold in terms

[*] PhD, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34 20156, Milan, Italy.

[†] PhD Student, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34 20156, Milan, Italy.

[‡] Associate Professor, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34 20156, Milan, Italy.

[§] Associate Professor, Department of Aerospace Science and Technology, Politecnico di Milano, Via La Masa 34 20156, Milan, Italy.

of accepted collision probability level.[5] Moreover, the accuracy of the computed Probability of collision (PoC) strongly impacts decision making by mitigating collision risk or by avoiding unnecessary Collision Avoidance Maneuvers (CAM) execution.

In light of these considerations it becomes a pressing issue to accurately quantify state uncertainties under the effect of low-thrust noise in a timely manner. In particular, the adoption of continuous thrust for long arcs may strongly affect the propagation of the statistics with respect to a pure ballistic motion. Hence, it is paramount to select an uncertainty propagation strategy that can deal with the actual performance of the thrusters, managing high-frequency thrust variations, thrust interruptions, and uncertain command execution times. In fact, these effects, coupled with the nonlinear dynamics and the orbital uncertainties mapping, sway the validity of the standard Gaussianity assumption adopted by state-of-the-art methods. In order to address the most general and accurate case of uncertainty propagation, one can solve the problem that stems from the Itô stochastic differential equation[6]

$$d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{G}(t)d\boldsymbol{\beta} \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the random state vector, $\boldsymbol{\beta}\,(t) \in \mathbb{R}^m$ is an m-dimensional Brownian motion process with zero mean and covariance $\mathbf{Q}(t)$. The vector function $\mathbf{f}(\mathbf{x}, t)$ captures the deterministic part of the dynamics, and $\mathbf{G}(t)$ is an $n \times m$ matrix characterizing the diffusion. For a given dynamical system that satisfies the Itô stochastic differential equation, the probability density function time evolution can be described by the Fokker–Planck partial differential equation (FPE).[7] However, solving a FPE in orbital mechanics is a difficult task, due to its high dimensional state–space and its underlying highly nonlinear dynamics.[8] Alternatively, to retrieve a complete statistical description, one may carry out computationally expensive particle-type studies such as Monte Carlo (MC) simulation.[9] However both methods have a considerable computational cost, owing to the entire statistical description of a trajectory. To address this issue, many analytical or semi-analytical techniques for orbital uncertainty propagation have been developed in recent years.[10]

By assuming Gaussian uncertainties and by further performing a local linearization of the dynamics (LinCov)[11] or a linearization in an "average" sense (CADET),[12] linear methods can completely characterize the propagated distributions. While these formulations are simple and efficient, their accuracy drops off for highly nonlinear systems, long-duration uncertainty simulations, or large initial uncertainty.

Several nonlinear techniques have been proposed in literature to refine the uncertainty realism. Probabilistic methods infer statistical properties from the propagation of a deterministically selected (small) number of samples (e.g., Unscented Transform,[13] Cubature methods,[14] Polynomial Chaos expansion[15]). On the other hand, dynamics-based procedures such as the state transition tensors (STTs)[16] were designed for motion approximation through higher order Taylor series expansion.[17] However, this method may require a significant computational effort to extract the required partials for high-fidelity dynamic systems. Thanks to automatic differentiation provided by differential algebra (DA),[18] the disadvantages of STTs methods can be overcome.[19] Indeed, DA provides a flexible tool that can be leveraged in many different ways to propagate uncertainties with lessened computational burden: polynomial evaluations initial distributions (i.e., MC-like), Isserlis' formula (i.e., STT-like approach without the computational effort), linear propagation (i.e., LinCov-like).[19] Nonetheless, none of the described strategies can deal with the high-order uncertainty propagation with low-thrust noise in a timely manner. This is indeed a particularly daunting task as the introduction of process noise results in a massive (in principle infinite) increase in the number of uncertain variables and the need to assess their effect along the trajectory with ad-hoc methods.[20] Typically, thrust noise is accounted for a covariance inflation term after propagation. However, this approach disregards any effect of noise on higher order moments of the statistics, as well as any coupling between uncertainty in the initial state and in the thrust. In addition, even by assuming perfect propagation of the uncertainty, any nonlinear transformation warps the Gaussian shape of a distribution; hence, it becomes necessary to provide a different description of the uncertainty. A possible solution is to describe the final uncertainty with Gaussian mixture models (GMM).[21–23] The key concept behind GMM is to separate a large domain into many smaller subdomains, represented with Gaussian mixture elements (GMEs), on which the dynamics can be linearized. The weights combining GMEs can then be determined both statically[22] as well as dynamically during propagation.[21]

To tackle all of these challenging aspects while providing a computationally efficient approach, this work proposes two separate methods which combine accurate nonlinear techniques with their computationally light linearized counterparts. Namely, the envisioned approaches rely on carefully mixing GMM with DA. Indeed, the bounty of the combination of DA with GMM methods has also been exploited in the past for problems without low-thrust.[24,25] The particular choice of GMM can also be explained by looking at its adaptability to PoC computation in long-term encounters[23] without requiring additional computational effort. The first technique is called Adaptive Differential Algebraic Gaussian Mixture Model (A-DAGMMo), and it constitutes a DA-based propagation scheme that describes the uncertainty evolution with a GMM. The peculiarity of this technique is that a nonlinearity index is computed during the propagation of each GME, effective to understand whether it is necessary to adaptively split the element to avoid the onset of nonlinearity. Besides providing automatic partials (which are useful for nonlinearity and split computations), DA also provides the state transition maps necessary to include the thrust noise during propagation. The second proposed method is the Stochastic Taylor Model (STyMo): a DA extension that endows the ODE flow expansion of the final state with an expansion of the embedded covariance inflation linked to the process noise. After propagating these two polynomial maps the final distribution of states can be retrieved. A GMM can then be fit to this distribution and the polynomial map of the covariance inflation is evaluated at the location of the GMM's means to retrieve the covariance correction term. By combining these two components, the final probability distribution is obtained as a GMM.

After a brief introduction on the mathematical tools provided by Differential Algebra, the next section will be dedicated to a thorough description of the developed methods. Subsequently, a following section will be dedicated to the description of the study cases. Furthermore, this work will present a numerical assessment of STyMo and A-DAGMMo in terms of both accuracy of the uncertainty propagation as well as on their usage and effect on PoC computation. To conclude, a final section will be dedicated to the discussion of results and to propose a way forward.

DYNAMICS-BASED UNCERTAINTY PROPAGATION

This section presents the description of the two dynamics-based UP advanced methods proposed in this work. Such methods combine the benefits of automatic differentiation native to DA techniques with the capability to deal with simplified linearized models typical of GMM. Before diving into the specifics of each method, it is necessary to provide a brief introduction to DA and its benefits.

Expansion of ODE flow with DA

DA brings the treatment of functions and the operations on them to the computer environment in a similar way as the treatment of real numbers as presented in.[18] The implementation of DA in a computer allows to compute the Taylor coefficients of a function up to a specified order k in a fixed amount of effort, and it can be used to perform composition of functions, to invert them, to solve nonlinear systems explicitly, to differentiate, and to integrate.
Given a general set of ODEs described by:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}) \tag{2}$$

The main advantage of DA is that, by setting the initial state as a DA variable and by carrying out all the evaluations in the DA environment, it is possible to retrieve the arbitrary order expansion of the flow of such set of ODEs with respect to variations of the initial condition as

$$[\mathbf{x}_f] = \mathbf{x}_f + \mathcal{T}^N_{\mathbf{x}_f}(\boldsymbol{\delta}\mathbf{x}_0) \tag{3}$$

After integrating the ODEs, the result of the final step of integration is a constant coefficient of the nominal propagated state $\mathbf{x}_f$ plus a N-th order Taylor expansion of the flow expressed in terms of variation of the initial state $\boldsymbol{\delta}\mathbf{x}_0$ at the final time t_f: $\mathcal{T}^k_{\mathbf{x}_f}(\boldsymbol{\delta}\mathbf{x}_0)$. DA techniques were developed for the ESA in the C++ DACE library by Dinamica Srl.[26]

Adaptive Differential Algebraic Gaussian Mixture Model

This approach assumes that a Gaussian distribution remains Gaussian under a linear transformation and that any regular nonlinear transformation behaves almost linearly if the domain of interest is sufficiently small. Suppose to have a Gaussian distribution at an initial time t_0, owing to a non-linear dynamics propagation the covariance is inflated and distorted over time. To overcome this issue, the A-DAGMMo method is conceived to split the original GME into an increasing number of GMMs as the dynamics are forwardly propagated through a DA based nonlinearity index. The routine is organized in this way:

1. Initialization of the GMM at time instant t_0, this case also includes the simple one of a single GME with weight 1 for the initial distribution, but also allows to propagate GMMs directly.

2. Each GME is added to a list of elements that need to be propagated until t_f and the propagation module starts to propagate each of them.

3. The propagation of mean and covariance is carried out both at first and at second order and at each step a nonlinearity metric is computed to detect the onset of nonlinearity (i.e., the need to further split uncertainty).

4. If the propagation threshold is not violated, the element is propagated until the end of the time window. Otherwise, a univariate splitting library[23] is applied to split the element in a set of new GMEs which are added to the list of elements that still need propagation, and the algorithm goes back to step 2.

As an additional safeguard, the algorithm is endowed with a minimum weight threshold for the covariances of the GMEs. If a single GME has a weight lower than this threshold it is not split anymore even if it violates the nonlinearity condition. When the propagation finishes for all GMEs, the final GMM can be assembled. The following subsections provide additional details on some of the key steps of this method, while a scheme of A-DAGMMo is reported in Figure 1.

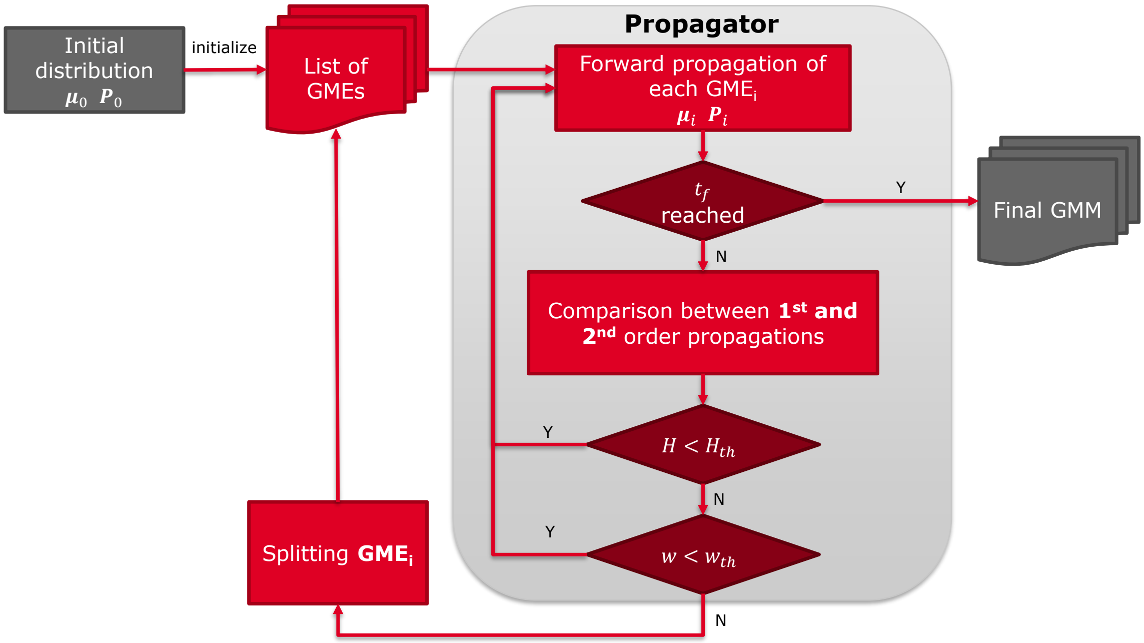


Figure 1. Schematics of the A-DAGMMo approach.

Multi-Order Mean and Covariance Propagation By calling t_0 the initial propagation time of a given GME, it is possible to characterize this distribution through an initial mean $\mathbf{x}_0$, covariance $\mathbf{P}_0$, and weight w_0. If the mean state is initialized as a DA variable and if the numerical integration of the ODEs in Eq. (2) are carried out in a DA environment, then the ODE flow expansion as in Eq. (3)) at each time instant can be obtained. For the purpose of this study, the ODE integration is carried out with a 7th order Runge Kutta numerical scheme with 8th order error control to adapt the step size during propagation. Moreover, the maximum order of the polynomial expansion for this method is set to $N = 2$. Indeed, this number should be enough to capture the incipient nonlinearity during propagation while not being too computationally demanding. After integration of these equations, a DA polynomial map for the final state will therefore be available up to second order. By truncating this polynomial expansion to first order the state transition matrix up to a certain time step t_k (i.e., $\mathbf{\Phi}(t_k)$) is retrieved and the noiseless covariance can be propagated by adopting the LinCov technique[11] as in Eqs. (4 – 6):

$$\mathcal{T}^1_{\mathbf{x}_k}(\boldsymbol{\delta}\mathbf{x}_0) = \mathbf{\Phi}(t_k) \tag{4}$$

$$\hat{\mathbf{x}}_k = \mathbf{\Phi}(t_k)\mathbf{x}_0 \tag{5}$$

$$\hat{\mathbf{P}}_k = \mathbf{\Phi}(t_k)\mathbf{P}_0\mathbf{\Phi}^\top(t_k) \tag{6}$$

with the superscript 1 indicating the truncation order of the polynomial map, whereas the accent ˆ has been used to characterize means and covariances obtained with a linear propagation. On the other hand, the second-order propagation[19] leverages the Isserlis' formula with maximum order $N = 2$, and is instead indicated with the ˜ accent:

$$\tilde{x}_{k,i} = \sum_{p_0+p_1+\cdots+p_{n-1}\leq N} c_{\tilde{x}_{k,i}(p_0,p_1,\ldots,p_{n-1})} E\left\{\delta x^{p_0}_{0,0}, \delta x^{p_1}_{0,1}, \ldots, \delta x^{p_{n-1}}_{0,n-1}\right\} \tag{7}$$

$$\tilde{P}_{k,i,j} = \sum_{p_0+p_1+\cdots+p_{n-1}\leq 2N} c_{\tilde{P}_{k,i,j}(p_0,p_1,\ldots,p_{n-1})} E\left\{\delta x^{p_0}_{0,0}, \delta x^{p_1}_{0,1}, \ldots, \delta x^{p_{n-1}}_{0,n-1}\right\} \tag{8}$$

Where i and j stand for the corresponding components of the arrays for the mean $\tilde{\mathbf{x}}_k$ and the covariance $\tilde{\mathbf{P}}_k$ at time t_k. In these expressions, the coefficients $c_{A_{k,i,j}(p_0,p_1,\ldots,p_{n-1})}$ indicate the coefficient of the expansion of the i, j component of the generic quantity A at time t_k corresponding to the ordered combination of variable exponents for the n dimensional state space $[p_0, p_1, p_2, \ldots, p_{n-1}]$.

The two distributions are compared at t_k according to the Hellinger Distance to determine the need to split the current Gaussian distribution:[27]

$$H^2 = 1 - \frac{\det\left(\hat{\mathbf{P}}_k\right)^{\frac{1}{4}} \det\left(\tilde{\mathbf{P}}_k\right)^{\frac{1}{4}}}{\det\left(\frac{\hat{\mathbf{P}}_k+\tilde{\mathbf{P}}_k}{2}\right)^{\frac{1}{2}}} \exp\left(-\frac{1}{8}\left(\hat{\mathbf{x}}_k - \tilde{\mathbf{x}}_k\right)^\top \left(\frac{\hat{\mathbf{P}}_k + \tilde{\mathbf{P}}_k}{2}\right)^{-1} \left(\hat{\mathbf{x}}_k - \tilde{\mathbf{x}}_k\right)\right) \tag{9}$$

This quantity expresses the distance between the two distributions and can therefore inform about the onset of nonlinearity during propagation. Moreover, it has the interesting property of being bounded between 0 and 1, which allows setting nonlinearity thresholds that should hold true for varying test cases.

This metric only accounts for the bulk of the nonlinearity coming from the deterministic part of the dynamics. Therefore, A-DAGMMo needs to be further adapted to account for the covariance inflation term owed to the presence of thrust noise. This is done as a one step correction of the covariance when the GME reaches its final propagation time: this can happen if the nonlinearity threshold is violated as well as if the final propagation time is reached. Indeed, if H^2 is less than a given threshold H^2_{th} there is no need to further split the GME. In this case, the propagation continues to time step t_{k+1} and the procedure is repeated until eventually the final time of the propagation is reached. The final mean and covariance of the GME are then computed with the second-order expansion of the DA and they are inflated with a term accounting for thrust noise. The computation of this inflation term from the initial time of the propagation t_0 until the final time t_f can be obtained as:[12]

$$\mathbf{\Delta P} = \int_{t_0}^{t_f} \mathbf{\Phi}(\tau)\mathbf{G}(\tau)\mathbf{Q}\mathbf{G}^\top(\tau)\mathbf{\Phi}^\top(\tau)d\tau \tag{10}$$

In this equation the term $\mathbf{Q}$ is adopted to describe the covariance of the process noise $\mathbf{w}$. Moreover, the state transition matrix $\mathbf{\Phi}(\tau)$ is retrieved automatically from the DA expansion of the flow. Similarly, $\mathbf{G}(\tau)$ is given by the gradient of the dynamics with respect to the input perturbation $\mathbf{G}(\tau) = \partial \mathbf{f} / \partial \mathbf{w}$ evaluated about the nominal trajectory. Also in this case the computation comes easy thanks to automatic differentiation. To conclude, the integration is performed through a simple numerical scheme (trapezoidal integration). The final covariance for each propagated GME will be therefore given as:

$$\mathbf{P}_f = \tilde{\mathbf{P}}_f + \mathbf{\Delta P} \tag{11}$$

On the other hand, if during the propagation the Hellinger distance threshold is violated, the propagation of the GME is stopped, the GME is split in a GMM, and each of the new components is corrected for covariance inflation from the beginning of their propagation up to the achieved splitting instant. More details on this approach are provided in the following subsection.

Adaptive Split Procedure This subsection describes the procedure that is carried out for each GME once the nonlinearity threshold becomes breached during propagation at timestep t_*. Before proceeding any further, another check is made in order to limit the number of total splits. In fact, a minimum weight w_{th} is imposed on the GMM, and should a GME with a weight lower than this threshold violate the nonlinearity constraint, it would still be propagated without further split. On the contrary, if it is possible to split the current GME, the direction in which this has to be split is determined thanks to a state-of-the-art algorithm.[28] First of all the eigenvectors of the covariance at the beginning of the propagation of the current GME $\mathbf{P}_0$ are determined and here reported with the notation $\mathbf{z}_0^i$ to indicate the i-th eigenvector of the matrix at initial time. For each of these eigenvectors, the nonlinearity of the ODE flow is computed as

$$\phi^i = \frac{\|\mathbf{J}(\mathbf{z}_0^i) - \mathbf{J}(\mathbf{x}_0)\|}{\|\mathbf{J}(\mathbf{x}_0)\|} \tag{12}$$

$$\mathbf{J}(\mathbf{x}) = \left. \frac{\partial \left[\mathbf{x}_*\right]}{\partial \boldsymbol{\delta} \mathbf{x}_0} \right|_{\mathbf{x}} \tag{13}$$

The nonlinearity index ϕ^i simply quantifies the variation of the Jacobian of the ODE flow $\mathbf{J}$ from the center of the initial distribution and it clearly goes to 0 in case of a linear propagation. In particular, the Jacobian $\mathbf{J}(\mathbf{x})$ expresses the variation of the ODE flow at time t_* for a variation of the initial state of the propagation. This quantity can be evaluated easily thanks to automatic differentiation available in DA. Once the eigenvector with the largest index ϕ^i has been determined, it is used as a split direction $\mathbf{d}$. By knowing the number of desired splits n_s, a univariate splitting library[23] is applied in the $\mathbf{d}$ direction. The outcome of this sub-routine is a new set of n_s means $\left[\mathbf{x}_0^0, \ldots, \mathbf{x}_0^{n_s-1}\right]$, covariances $\left[\mathbf{P}_0^0, \ldots, \mathbf{P}_0^{n_s-1}\right]$, and weights $\left[w_0^0, \ldots, w_0^{n_s-1}\right]$ at the initial time of the propagation t_0 for the current GME that needed splitting. Thanks to the polynomial map $\left[\mathbf{x}_*\right](\boldsymbol{\delta}\mathbf{x}_0)$ it is also possible to propagate each of the newly generated GMEs directly up to t_* without having to integrate each one of them numerically. To do so, the map is re-expanded about each new mean obtained from the split by substituting $\left[\mathbf{x}_*\right](\mathbf{x}_0^s - \mathbf{x}_0 + \boldsymbol{\delta}\mathbf{x}_0^s)$ for each $s = 0, \ldots, n_s - 1$. The newly expanded maps can be used with Eqs. (7 – 8) to retrieve the propagated means and covariances. Moreover, these new expansions can also be exploited in Eq. (11) to compute the covariance inflation terms appropriately for each of the new GMEs going from t_0 to t_*. At the end of the split procedure, the new set of GMEs at t_* is added to the list of GMEs that need to be propagated and their starting time is set to be t_0 for the next propagation. The total number of GMEs cumulated during the automatic split procedure will grow exponentially with the number of splits, hence, the computational effort can be limited by bounded by tuning the number of GMEs for each split and the split threshold.

Stochastic Taylor Model

The second method proposed extends DA to include the possibility of handling process noise in uncertainty propagation. The key idea of the STyMo approach is to obtain a polynomial expansion of the ODE flow for the state which is endowed with a linearized correction term for the covariance to account for the thrust noise. The overall procedure can be summed up as a sequence of steps:

1. The initial state is set as a DA variable and its ODEs are propagated to retrieve an higher-order expansion of the flow with respect to the initial state variation. This system of ODEs is coupled with additional equations accounting for the linearized correction of the covariance. This coupling allows to retrieve the covariance inflation term as an expansion of the initial condition, therefore it can be used to re-expand this correction term about any trajectory stemming from an initial condition in the neighborhood of the initial state.

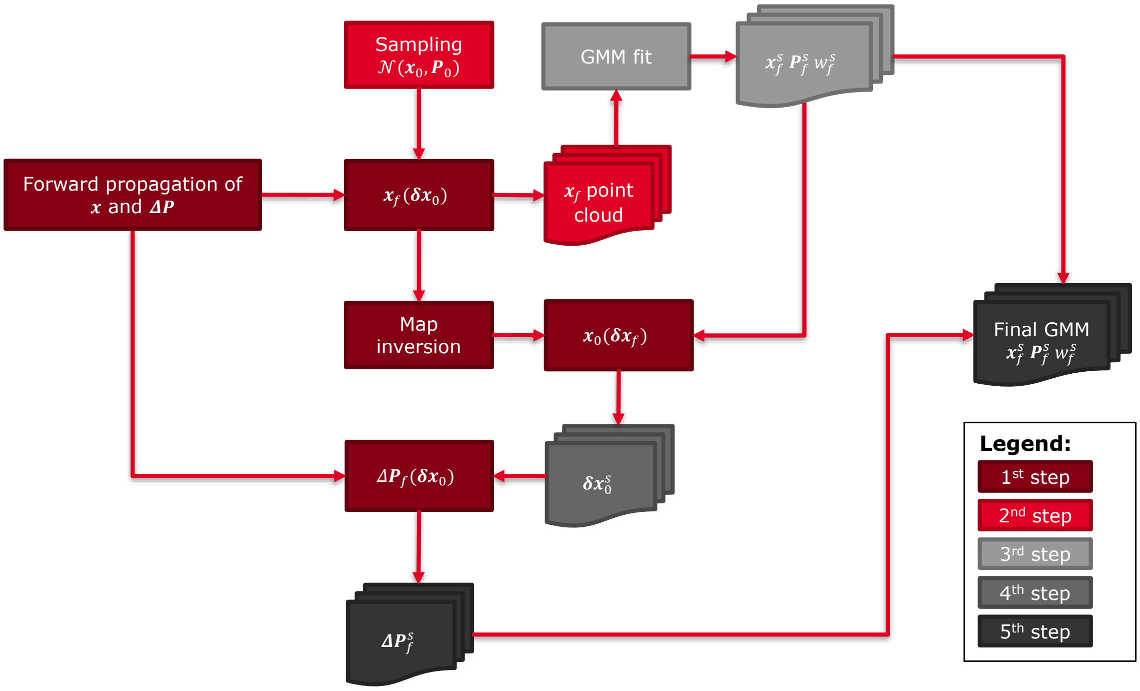
2. After propagating these two polynomial maps a DA Monte Carlo simulation is performed on the state map to retrieve the final distribution of states.

3. A GMM is then fit to this point cloud so that a set of means, weights, and covariances can be used to summarize the final distribution.

4. The initial state that generates each mean of the final GMM is obtained through a polynomial map inversion of the ODE flow of the state.

5. Then, these initial conditions can be used to evaluate the polynomial map of the covariance inflation. This term is simply summed to the final covariance of each GME without modifying its weight.

An overall description of the STyMo process is provided in Figure 2 whereas additional details on the covariance inflation flow expansion and its exploitation are better detailed in the following subsections.



Sampling $\mathcal{N}(\boldsymbol{x}_0, \boldsymbol{P}_0)$
GMM fit
$\boldsymbol{x}_f^s\ \boldsymbol{P}_f^s\ w_f^s$
Forward propagation of $\boldsymbol{x}$ and $\boldsymbol{\Delta P}$
$\boldsymbol{x}_f(\boldsymbol{\delta x}_0)$
$\boldsymbol{x}_f$ point cloud
Map inversion
$\boldsymbol{x}_0(\boldsymbol{\delta x}_f)$
Final GMM $\boldsymbol{x}_f^s\ \boldsymbol{P}_f^s\ w_f^s$
$\boldsymbol{\Delta P}_f(\boldsymbol{\delta x}_0)$
$\boldsymbol{\delta x}_0^s$
$\boldsymbol{\Delta P}_f^s$
Legend:
1st step
2nd step
3rd step
4th step
5th step

Figure 2. Schematics of the STyMo approach.

Covariance Inflation Expansion The initial state is hence set as a DA variable and the following set of equations is propagated from the initial conditions $[\mathbf{x}_0] = \mathbf{x}_0 + \boldsymbol{\delta}\mathbf{x}_0$ and $\mathbf{\Delta P}_0 = \mathbf{0}$:

$$\dot{\mathbf{x}} = \mathbf{f}\left(t, \mathbf{x}, \mathbf{u}\right) \tag{14}$$

$$\dot{\mathbf{\Delta P}} = \mathbf{F}\mathbf{\Delta P} + \mathbf{\Delta P}\mathbf{F}^\top + \mathbf{G}\mathbf{Q}\mathbf{G}^\top \tag{15}$$

Having adopted the notation $\mathbf{G}$ and $\mathbf{F}$ to characterize the following partial derivatives about the nominal trajectory given by:

$$\mathbf{F} = \frac{\partial \mathbf{f}}{\mathbf{x}} \tag{16}$$

$$\mathbf{G} = \frac{\partial \mathbf{f}}{\mathbf{u}} \tag{17}$$

Since the nominal trajectory is a polynomial expansion in $\boldsymbol{\delta}\mathbf{x}_0$, through the dependency from $\mathbf{F}$ and $\mathbf{G}$ also $\boldsymbol{\Delta}\mathbf{P}$ is retrieved as a polynomial expansion about the initial state. Therefore, at the desired final time of the propagation two polynomial expansions will be available, one for the state such as in Eq. (3) and one for the covariance inflation:

$$[\boldsymbol{\Delta}\mathbf{P}_f] = \boldsymbol{\Delta}\mathbf{P}_f + \mathcal{T}_{\boldsymbol{\Delta}\mathbf{P}_f}\left(\boldsymbol{\delta}\mathbf{x}_0\right) \tag{18}$$

The reasoning behind this approach is to propagate only the covariance inflation term owed to the presence of thrust noise in the region neighbouring the nominal trajectory. By adopting a polynomial expansion of the ODE flow, it becomes possible to re-expand the covariance inflation term about any trajectory sampled from the state transition map. Eq. (15) can be retrieved according to the following procedure that relies on propagating covariance with and without process noise:[12]

$$\dot{\overline{\mathbf{P}}} = \mathbf{F}\overline{\mathbf{P}} + \overline{\mathbf{P}}\mathbf{F}^\top + \mathbf{G}\mathbf{Q}\mathbf{G}^\top \tag{19}$$

$$\dot{\mathbf{P}} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^\top \tag{20}$$

By making the assumptions of local linearization,[12] the presence of thrust noise does not influence the nominal trajectory. As a consequence $\mathbf{F}$ and $\mathbf{G}$ will be the same for both set of ODEs. This consideration allows to directly propagate the inflation term owed to the presence of thrust noise by taking their difference while defining $\boldsymbol{\Delta}\mathbf{P} = \overline{\mathbf{P}} - \mathbf{P}$. These assumptions imply that the bulk of nonlinearity in the uncertainty propagation will depend on the unperturbed nonlinear motion and will be captured by the high-order expansion of the flow of ODEs for the mean. On the other hand, they also entail that the process noise will be account for a term that can be obtained via local linearization and that does not significantly impact the propagation of the mean state.

Evaluation of final Distribution The realization of the final state ODE flow expansion allows to exploit the DA Monte Carlo approach to propagate a large number of samples. These samples are drawn from the initial distribution and propagated through the polynomial map without having to numerically integrate each trajectory individually:[19,29]

$$\mathbf{x}_f^n = \mathbf{x}_f + \mathcal{T}_{\mathbf{x}_f}\left(\mathbf{x}_0^n - \mathbf{x}_0\right) \tag{21}$$

with $\mathbf{x}_0^n$ and $\mathbf{x}_f^n$ being the initial and final state of the n-th sample.
This procedure produces a point cloud at t_f, which can be fitted with an arbitrary n_s number of GME elements*. Each GME has a specific mean $\mathbf{x}_f^s$, covariance $\mathbf{P}_f^s$ and weight w_f^s. The higher the weight, the more the linked GME contributes to the Probability Density Function (PDF) at t_f. Up to this point, the GMEs do not take into account thrust noise covariance. To include this contribution, the polynomial map is exploited a second time. In particular, by exploiting polynomial map inversion and evaluating it for each mean of the final GME

$$\boldsymbol{\delta}\mathbf{x}_0^s = \mathcal{T}_{\mathbf{x}_f}^{-1}\left(\mathbf{x}_f^s - \mathbf{x}_f\right) \tag{22}$$

it becomes possible to retrieve the variation from the initial condition that generated the mean of the current GME under investigation. This initial variation can be used in the evaluation of the covariance inflation term for each mean of the GME by evaluating Eq. (18). Once $\boldsymbol{\Delta}\mathbf{P}_f^s$ has been evaluated for every GME, the covariance inflation term is simply summed to the final covariance of the mixand $\mathbf{P}_f^s$.

*https://scikit-learn.org/0.17/modules/generated/sklearn.mixture.GMM.html

NUMERICAL SIMULATIONS

This section provides a description of the analyzed test cases. For both proposed methods, the dynamics of the systems are assumed to be defined by:

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u} + \mathbf{w}) = \begin{cases} \dot{\mathbf{r}} = \mathbf{v} \\ \dot{\mathbf{v}} = -\frac{\mu}{r^3}\mathbf{r} + \frac{1}{2}\rho(r)\left(\frac{C_d A}{m}\right)v^2 + \mathbf{u} + \mathbf{w} \end{cases} \tag{23}$$

The state is here represented by position and velocity $\mathbf{x} = [\mathbf{r}; \mathbf{v}]$, and it is further assumed that any given control history can be parametrized with a function of time $\mathbf{u}$ plus additive process noise $\mathbf{w}$. In this work the dynamics is assumed to be constituted by Keplerian acceleration with gravitational parameter μ, a simplified drag model, and the thruster acceleration (with its noise). Concerning the drag model, the ballistic coefficient is given by the ratio of front area A times the drag coefficient C_d and it is divided by the mass of the object m. To conclude, the atmospheric density ρ is assumed to be a function of the altitude only. The following subsections will briefly introduce additional details on how the atmospheric density and process noise are modelled, before introducing the analyzed orbital test cases.

Density Model

The implemented atmospheric density model needs to cover altitudes of at least 1300 km to provide drag values for all envisioned test cases in LEO. Since a simple exponential model would not accurately account for molecular dissociation happening above 80 km, the implemented simplified model relies on the Jacchia77 empirical model,[30] which provides density profiles up to 2500 km. However, this model requires thermospheric temperatures as an input, which would mean having a different value depending on magnetic indexes and by the solar flux: all quantities which are influenced by the time of day, the day of the year, and year in the solar cycle. To simplify this estimate by reducing it to only a dependence on altitude, this input thermospheric temperature is estimated with the NRLMSISE-00[31] atmospheric model at the 2022 spring equinox, while using magnetic flux indices which are commonly adopted for long term predictions of space weather. Since the temperature distribution for a given altitude is not uniform (i.e. it depends on the relative position to the sun), also the atmospheric density at different latitudes and longitudes will be different depending on the location. To solve this final issue, these densities are averaged at different altitudes levels. The average values and their variation from the lower and upper bounds at each altitude are reported on the left of Figure 3. The averaged values can finally be interpolated with a function $\rho(r)$ that causes an estimation error always less than 2% for the altitude ranges achieved in the envisioned scenarios as illustrated on the right of Figure 3.

Process Noise Physical Model

To determine a physical model of the thrust noise covariance (i.e., the process noise covariance $\mathbf{Q}$) it is assumed that this can be modelled as a pointing and amplitude error. The process of determining this quantity starts by calling the reference thruster acceleration at time t as $\mathbf{u}_r = u_r \boldsymbol{\tau}_r$, where u_r indicates the desired amplitude while $\boldsymbol{\tau}_r$ represents the reference direction. This latter can be expressed in spherical coordinates thanks to the angles α and β:

$$\boldsymbol{\tau}_r = \cos\alpha\cos\beta\mathbf{i} + \sin\alpha\sin\beta\mathbf{j} + \sin\beta\mathbf{k} \tag{24}$$

By further assuming that error on pointing are small and Gaussian, we will have a perturbed pointing direction as $\boldsymbol{\tau} = \boldsymbol{\tau}_r + \boldsymbol{\Delta\tau}$. This perturbation is linear in the small angle variation $\boldsymbol{\Delta}\mathbf{y} = [\delta\alpha, \delta\beta]^\top$ as:

$$\boldsymbol{\Delta\tau} = \mathbf{R}\boldsymbol{\Delta}\mathbf{y} = \begin{bmatrix} -\cos\beta\sin\alpha & \cos\alpha\sin\beta \\ \cos\alpha\cos\beta & -\sin\alpha\sin\beta \\ 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \delta\alpha \\ \delta\beta \end{bmatrix} \tag{25}$$

Moreover, by assuming a small Gaussian perturbation on the amplitude, both effects can be superimposed.

$$\mathbf{u} = \mathbf{u}_r + \delta u \boldsymbol{\tau}_r + u_r \mathbf{R}\boldsymbol{\Delta}\mathbf{y} \tag{26}$$

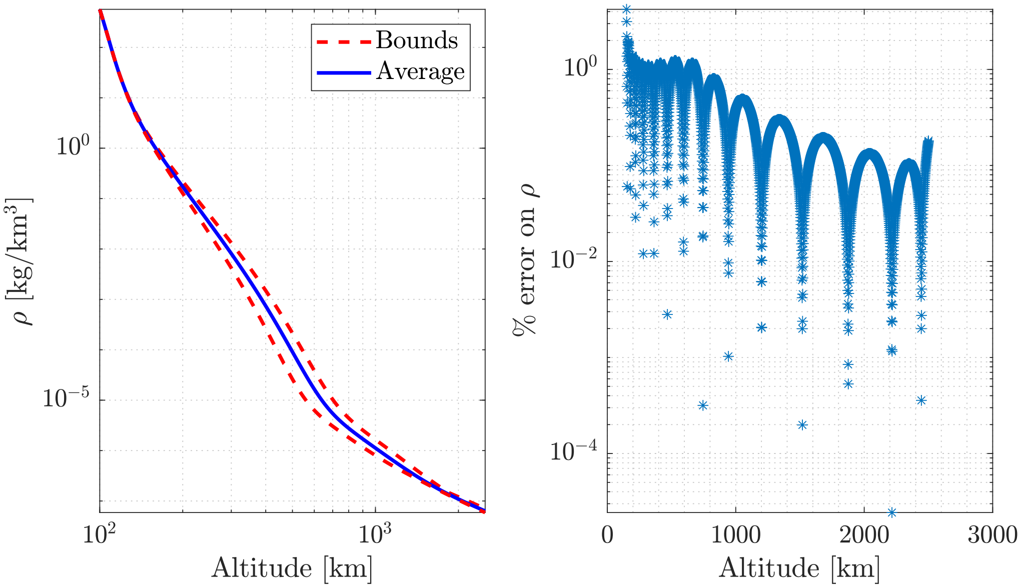


Figure 3. On the left, density estimation and upper/lower bounds at each altitude, on the right the error caused by interpolation.

To conclude, by assuming that the errors on amplitude and pointing are uncorrelated and have null mean, the covariance $\mathbf{Q}$ of the process noise disturbance $\mathbf{w} = \delta u \boldsymbol{\tau}_r + u_r \mathbf{R} \boldsymbol{\Delta} \mathbf{y}$ can be expressed as a combination of the covariances of amplitude error (σ_u^2) and pointing error ($\boldsymbol{\Sigma}_\mathbf{y}$).

$$\mathbf{Q} = \sigma_u^2 \boldsymbol{\tau}_r \boldsymbol{\tau}_r^\top + u_r^2 \mathbf{R} \boldsymbol{\Sigma}_\mathbf{y} \mathbf{R}^\top \tag{27}$$

For the envisioned test cases, 4 different combination of increasing uncertainties are adopted and their values are reported in Table 1.

Table 1. Summary of noise levels studied for the presented test cases

Noise Level	Amplitude error [%]	Angular error [deg]
Accurate	0.5	1.0
Average	1.0	2.0
Low	2.5	5.0
Inaccurate	5.0	10.0

Test Cases Description

The test cases have been provided with the collaboration of GMV. they include significant scenarios for long-thrusting (e.g., Orbit Raising) and orbit maintenance. Among the operational scenarios involving long thrusting, at least the following ones are considered:

- Low-LEO-to-LEO electric orbit raising, as currently carried out for instance by One-web satellites from around 500 km to around 1200 km following a spiral-like trajectory lasting months.
- GTO-to-GEO electric orbit raising, as currently carried out by GEO satellites equipped with low-thrust devices in order to raise the satellite from GTO orbit to the final GEO orbit.
- LEO disposal, where a LEO satellite orbit is lowered with low-thrust means at its end-of-life in order to comply with the guideline of 25 years maximum orbital lifetime.

- LEO Station Keeping (SK), as currently done by various types of satellites (communications, Earth-observation, etc.) following different strategies (ground-track control, constellation control).
- GEO orbit maintenance, where GEO satellites with electric propulsion carry out daily low-thrust manoeuvres to compensate the orbital disturbances caused by Sun and Moon and the Earth geopotential.

For each of these scenarios, a reference orbit is generated, considering the typical constraints affecting low-thrust devices (e.g. eclipses, maximum slew rates, etc.). These reference orbits are then used to simulate an operational-like situation where measurements are available from the sensors being used, through the simulation of the observations and the subsequent orbit determination process, throughout different phases of the reference orbit. This allows to generate realistic orbital states and covariances, at the time of the last observation. The following sensor networks are considered for the observation simulation process, together with the expected visibility windows:

- On-board GNSS receivers, both below and above GNSS altitude, considering the typical size of primary and secondary lobes of the GNSS signal.
- Ranging stations, used to measure the slant range of the satellite with respect to a network of stations, whose number depends on the orbital regime considered (e.g. two in GEO).
- Space surveillance networks, composed of survey and tracking radars (typically for LEO) and telescopes (typically for MEO and GEO).

The a-posteriori realistic covariances are then used as input to perform the uncertainty propagation using the different approaches to be compared, including the uncertainty of the low-thrust propulsion system. A summary of the study cases is reported in Table 2.

Table 2. Operational scenarios investigated.

Scenario	Orbital Regime	Propagation Time [days]	Notes
LEO to LEO	LEO (500 km)	7	Earth Orbit Raising with continuous thrust
LEO DISPOSAL	Low LEO (Starlink)	7	End of disposal for Starlink-like satellite with continuous thrust
LEO SK	Low LEO (Starlink)	3	Daily along-track short maneuvers
GTO to GEO	GTO	15	Beginning of Orbit Raising (i.e., near GTO) with continuous thrust
GEO SK	GEO	15	Daily maneuvers of long duration

Additionally, in order to assess the suitability of the selected approach for characterizing conjunctions, collisions are also simulated as part of this process with the estimated orbits of the operational satellites (i.e. primary object) and a background population of space debris objects. The simulated conjunctions are reported in Table 3.

Table 3. Analyzed Conjunctions.

Scenario	Relative Speed (m/s)	Conjunction Angle [deg]	Time to Conjunction [days]
LEO to LEO	1000	90	3
LEO DISPOSAL	15000	179	2
LEO SK	5000	40	4
GTO to GEO	16000	178	4
GEO SK	500	70	5

RESULTS

The accuracy of each scenario will be compared against a Monte Carlo simulation of 50000 samples, the metrics adopted will be:

- Runtime normalized with respect to benchmark MC (500 samples)
- L_2 norm of final mean position and velocity errors
- L_2 norm (i.e., Frobenius norm) of final position and velocity covariance errors
- Maximum Total Variation Distance (TVD) among all states
- Maximum Mean Integrated Square Error (MISE) among all states

These two latter measures have been introduced to account for the non-Gaussianity of the distribution. First, the MISE will be evaluated per each dimension and normalized with the distribution obtained for each dimension by the Monte Carlo analysis. Secondly, also the TVD will be analysed per each dimension. To conclude, only the maximum MISE and maximum TVD will be provided. While these latter metrics should provide some insight into the description of the non-Gaussianity, they only tell part of the story. Indeed, by considering each dimension on its own, all information related to the coupling between variables is lost. When looking at the computational time, each of the reported times have been normalized with the runtime of the corresponding Monte Carlo simulation of 500 samples to reduce the computational effort.

Hyperparameters Tuning

Both presented methods have hyperparameters that need to be tuned for proper execution. The STyMo has been run at expansion orders 2 and 3 for all scenarios. This choice has been dictated by the compromise between accuracy and computational time. In fact, despite the upgrade of the expansion order from 2 to 3 producing an average increase of $\sim$ 6 times in the computational time, some cases benefit from the higher-order. Indeed, changing from order 2 to 3 produced an average improvement of $\sim 10\%$ across all accuracy metrics in some of the tested cases (e.g., GTO to GEO). While moving to even higher orders has been taken into consideration, it has ultimately been decided to adopt order 3 for all simulations in light of the limited benefits and considering the high penalty in computational time in which the algorithm would incur. Moreover, the results obtained with order 3 proved to be sufficiently accurate for all the envisioned test cases. As a consequence, the only hyperparameter that will need to be defined for the STyMo is the number of GMEs chosen to represent the final distribution. To determine this number, an analysis can be carried out to determine the last number of splits that provides a significant improvement to the performance metrics. This analysis has been performed for all scenarios for the lowest level of thruster uncertainty reported in Table 1. Once selected, this number is also used for all other levels of process noise. In this paper only the LEO to LEO case is reported for conciseness. Notice how Figure 4 does not include the mean error as the mean is basically unchanged by the splitting procedure. It is clear that using more than 9 GMEs in this scenario would not provide any significant benefit. By repeating this analysis for all test cases one can fix the number of splits to be used.

Concerning the A-DAGMMo method, the hyperparameters that need to be set are the number of splits to be performed when the nonlinearity is detected and the nonlinearity threshold at which to perform said split. The number of splits at each splitting step does not affect accuracy as observed from several trial runs of this code. Therefore, it has been decided to set it to 3 for all simulations. This choice only implies that if the nonlinearity is larger, there will be a need for more splitting steps, producing a similar computational time penalty as using a higher number of splits per step and fewer splitting steps. In the application of this method, selecting a lower split produced a delay in the accumulation of the GMEs which granted a practical speed-up of the code in its current implementation. The approach for tuning the splitting threshold (i.e., the only remaining parameter) is to gradually decreasing this value trying to balance the computational effort and the accuracy. Also in this case the tuning is performed on the lowest level of thruster noise and the thusly

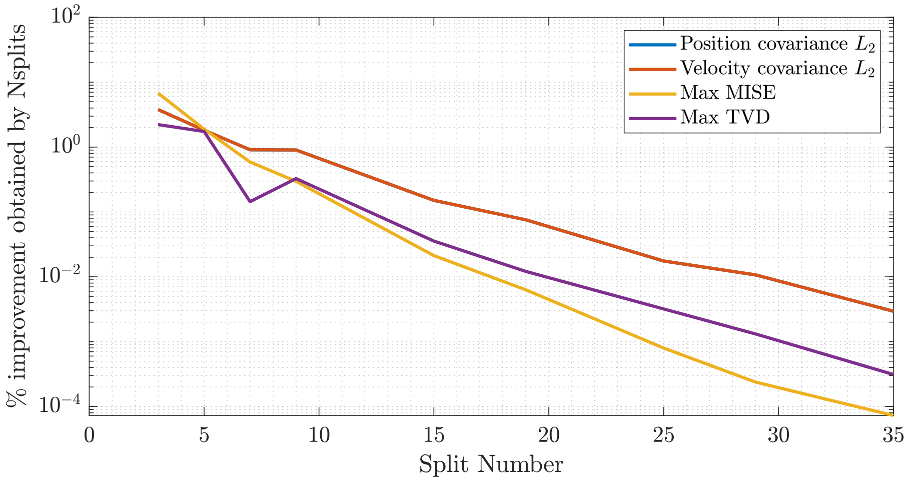


Figure 4. Improvement obtained across performance metrics for increasing number of splits in the LEO to LEO scenario.

obtained parameters are kept constant for each scenario. For the LEO to LEO scenario the best values of splitting threshold were found to be in $[0.01, 0.05]$.

Comparison of Uncertainty Propagation Approaches

As expected, both propagation methods show degrading performances for increasing duration of the propagation across all metrics as illustrated for the specific LEO to LEO case in Figure 5.

To account for this behaviour, while also considering the varying level of thruster error in Table 1, a cumulative index called Process Noise Index (PNI) is used to compare performances of different scenarios and algorithms. This PNI is computed as the ratio between initial and final sizes of the state covariances obtained from the benchmark MC analysis.

$$\text{PNI} = \frac{\det\left(\mathbf{P}_f\right)}{\det\left(\mathbf{P}_0\right)} \tag{28}$$

The results are reported in a series of summary graphs considering all combination of scenarios (i.e., with different markers) and algorithm (i.e., with different colors) when propagated at their final time as reported in Table 2. Each of these combinations will have a 4 different levels of PNI which are determined by the thruster noise performance.

A preliminary analysis of the results obtained after hyperparameter tuning evidenced some important results. In particular, it was observed that the hyperparameters of both methods can be fixed for similar scenarios. LEO to LEO and LEO DISPOSAL showed the same best value of GMEs for STyMo to be of 9 and a best nonlinearity threshold for A-DAGMMo in the range $[0.01, 0.05]$. However, these parameters may vary even for the same orbital regime if the propagation is particularly different, as in the case of the LEO SK where only a few, short maneuvers are taken: this case required 27 GMEs for STyMo and consistently retrieved a larger number of mixands (i.e., 17) for A-DAGMMo even though the threshold was set in the same range as the other LEO scenarios. GEO cases behave similarly with higher number of splits for STyMo (i.e., 15 for both) but with nonlinearity thresholds for A-DAGMMo of 1 order of magnitude (or lower for the GEO SK) smaller to obtain the same level of splits: this is also in line with the idea that GEO scenarios show overall lower nonlinearity levels.

Going into further details, Figure 6 reports the computational time for all combination of methods, scenarios, and thruster performance. It is possible to observe that STyMo is more costly in terms of computational

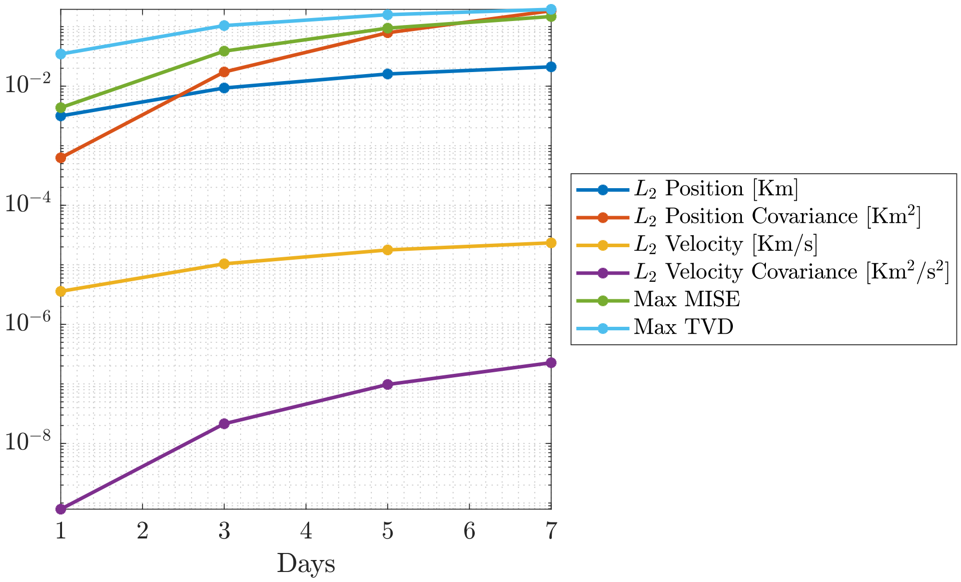


Figure 5. Performance degradation with duration of propagation in LEO to LEO scenario.

time, being it on average ~ 0.7 times the benchmark MC with 500 samples. However, its cost is not dependent on the accuracy of the thruster, nor with the level of nonlinearities: both of which contribute to an increase of PNI. On the contrary, A-DAGMMo shows a dependency of the computational time with PNI which is particularly relevant for some scenarios (e.g., LEO to LEO and LEO DISPOSAL). Despite this, it still is the faster method between the two, with an average normalized time across all PNI and scenarios of about 0.1. The increase of time for this method is to be expected for some challenging cases where the increasing nonlinearity makes the number of GMEs (adaptively determined) grow very rapidly for the hyperparameters selected with the lowest PNI. In this case in fact for the highest PNI the LEO to LEO required ~ 1000 GMEs, whereas the LEO DISPOSAL ended with 35 mixands.

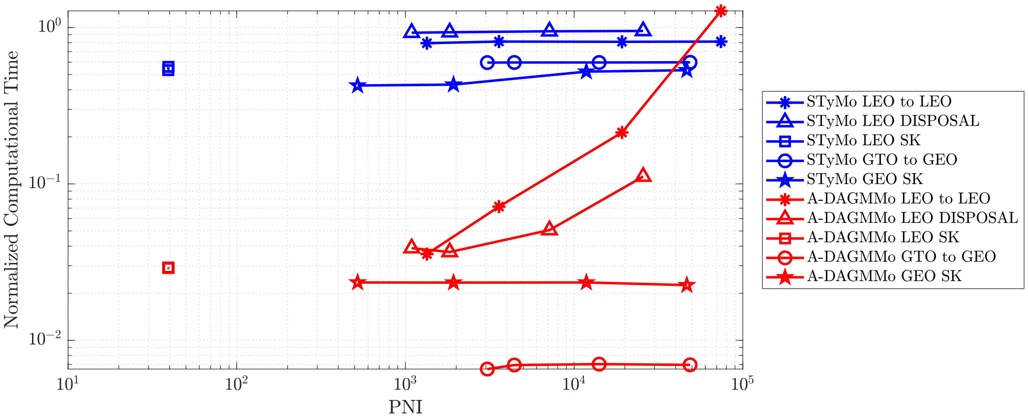


Figure 6. Normalized computational time for all combinations of scenario, method, and thruster uncertainty (i.e., increase in PNI).

Figure 7 shows the effect of PNI on the error of estimation of the mean. Both subfigures shows the same trends, with the estimation performed by either STyMo and A-DAGMMo not significantly influenced by an increase of PNI for each scenario. In addition, it can also be noticed how STyMo outperforms A-DAGMMo in the estimation of the mean. Nonetheless, both methods are affected by an increase in PNI for each scenarios when looking at the covariance error in Figure 8. In this case it is A-DAGMMo that takes the upper hand against STyMo in most scenarios.

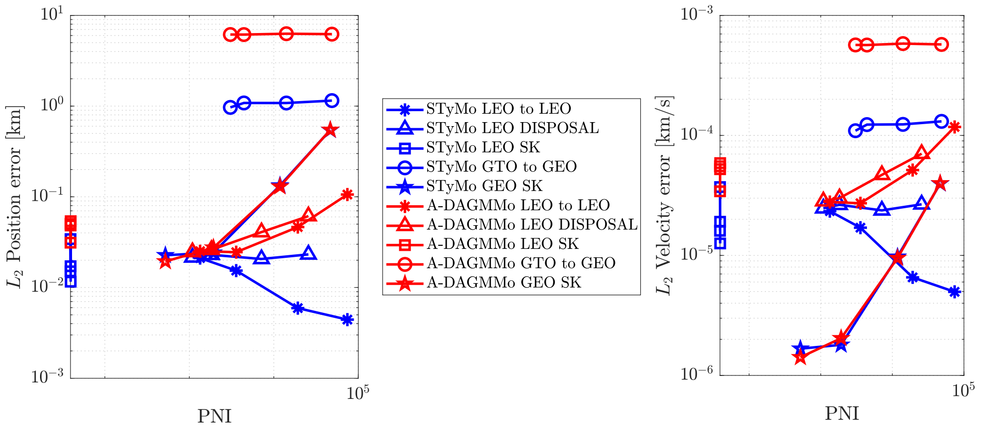


Figure 7. Error in the final mean position (left) and velocity (right) for all combinations of scenario, method, and thruster uncertainty (i.e., increase in PNI).

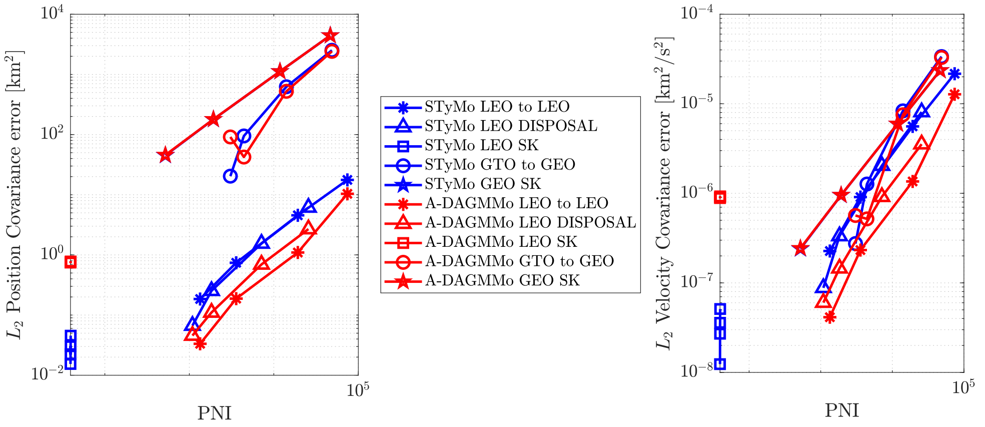


Figure 8. Error in the final covariance for position (left) and velocity (right) for all combinations of scenario, method, and thruster uncertainty (i.e., increase in PNI).

Since both methods are basically unchanged in terms of position but they accumulate a growing error in covariance, also the metrics capturing the non-Gaussian behaviours reflect similar trends in Figure 9. However, as explained during hyperparameter tuning, STyMo has the number of GMEs determined from a tradeoff analysis performed at the lowest level of PNI for each scenario. On the contrary, A-DAGMMo can still adapt it autonomously even if the parameters are fixed. This trend is clearly visible in the LEO to LEO and LEO DISPOSAL scenarios where the number of GMEs grow but they allow for a better representation of the uncertainty.

It is also worth mentioning that the LEO SK has very little variation of the PNI because it has the shorter propagation but also very short and few maneuvers. In this case as well as the GEO ones there is substantially no difference between the two approaches.

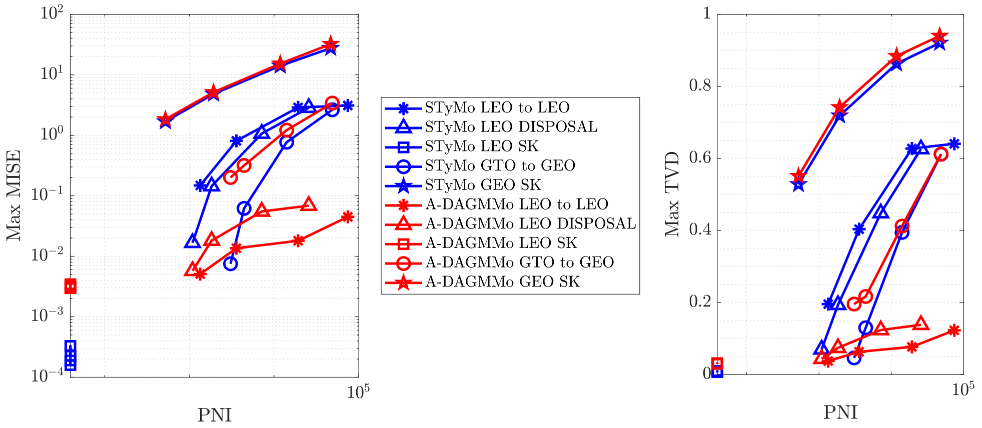


Figure 9. MISE (left) and TVD (right) for all combinations of scenario, method, and thruster uncertainty (i.e., increase in PNI).

Conjunction Analysis

Five conjunction scenarios described Table 3 have been analysed with the proposed uncertainty propagation methods. Each of the approaches produces two final GMMs to represent the final distributions at the estimated Time of Closest Approach (TCA) for primary and secondary objects. When both GMMs are available, the following method is adopted to compute the PoC:[23]

- Selecting a combination of a GME from the primary's GMM and a GME from the secondary's GMM.
- Searching for the correct TCA of the combination.
- Propagating locally the mean and covariance of both GMEs to the real TCA.
- Computing the PoC of the combination with a short-term encounter model such as Chan's model.[32]
- Summing all contributions from all possible combinations of GMEs according to Eq. (29).

$$\mathrm{PoC}_{\mathrm{GMM}} = \sum_{i=1}^{n_p}\sum_{j=1}^{n_s} w_i^p w_j^s \mathrm{PoC}_{\mathrm{Chan}}\left(\mathbf{x}_i^p\left(\mathrm{TCA}_{i,j}\right), \mathbf{x}_j^s\left(\mathrm{TCA}_{i,j}\right), \mathbf{P}_i^p\left(\mathrm{TCA}_{i,j}\right), \mathbf{P}_j^s\left(\mathrm{TCA}_{i,j}\right)\right) \tag{29}$$

This equation accounts for the weighted sum of n_p GMEs of the primary and n_s GMEs of the secondary. The weights of the GMEs are expressed as w^p, i for the i-th GME of the primary and w^s, j for the j-th GME of the secondary. Then, after computing the correct TCA of the combination of i and j mixands (i.e., $\mathrm{TCA}_{i,j}$), the new means and covariances of the GMEs can be propagated from the nominal TCA to the new value and they are reported as $\mathbf{x}_i^p\left(\mathrm{TCA}_{i,j}\right)$ and $\mathbf{P}_i^p\left(\mathrm{TCA}_{i,j}\right)$ for the primary and as as $\mathbf{x}_j^s\left(\mathrm{TCA}_{i,j}\right)$ and $\mathbf{P}_j^s\left(\mathrm{TCA}_{i,j}\right)$ for the secondary. This method implicitly assumes that PoC can be computed as a weighted sum of all contributions given by all combinations of GMEs for primary and secondary while assuming short term encounter. The summary of results is reported in Table 4.

The number of splits for STyMo has been selected after a trade-off analysis carried out on all scenarios. It was shown that (similarly to Figure 10) no significant change to the PoC happens after selecting 15 splits for both primary and secondary. In addition, the reference linear propagation case has always 1 GME for both primary and secondary by definition. On the other hand, A-DAGMMo determines this number automatically during propagation.

Table 4. PoC results for all scenarios given together with the number of GMEs for the primary (i.e., N_p) and secondary (i.e, N_s) objects.

Scenario	Linear Propagation *(N_p,N_s)*	STyMo *(N_p,N_s)*	A-DAGMMo *(N_p,N_s)*
LEO to LEO	7.8984E-05 *(1,1)*	7.9897E-05 *(3,5)*	7.8235E-05 *(15,15)*
LEO DISPOSAL	6.1197E-04 *(1,1)*	6.0673E-04 *(3,7)*	5.9741E-04 *(15,15)*
LEO SK	3.7338E-04 *(1,1)*	3.7008E-04 *(9,3)*	3.6334E-04 *(15,15)*
GTO to GEO	1.9472E-04 *(1,1)*	1.8967E-04 *(9,3)*	1.6375E-04 *(15,15)*
GEO SK	5.5866E-03 *(1,1)*	5.5866E-03 *(1,1)*	5.4691E-03 *(15,15)*

It is worth noticing how the number of splits of the secondary is the one that determines the largest variations of PoC for STyMo in the LEO to LEO scenario and that this behaviour is automatically captured by A-DAGMMo which uses a higher number of GMEs for the secondary object as illustrated in Table 4. Moreover, in the GEO SK scenarios nonlinearities are so small that no splits were determined by A-DAGMMo so that the same results as a local linearization are retrieved. Indeed, in this scenario the uncertainties are propagated for only 5 days. Conversely, in the previous uncertainty propagation assessment results were obtained after 15 days, which are enough to introduce enough nonlinearity to cause a split. For the same reasons, also STyMo retrieved very small variations of PoC for varying number of GMEs.

As Table 2 demonstrates, the PoC obtained with dynamics-based methods is always close to the value obtained through a linear propagation. This behaviour is expected as all the methods presented treat process noise with local linearizations built upon higher order information extracted from the flow of the dynamics.

Across all scenarios it is possible to observe an overall dilution of the PoC. This behavior however depends on many factors such as the weight of the particular pair of GMEs as well as the modification of the conjunction geometry and covariances owed to the local adjustment of the TCA. In any case, both STyMo and A-DAGMM provide consistent results across scenarios.

CONCLUSIONS

Both A-DAGMMo and STyMo methods retrieve consistent results. Different ways of including a covariance correction term based on local linearization output similar results. Despite being much faster than a MC analysis, the accuracy of these methods is not always excellent. One hypothesis is that this behavior is to be imputed to the process noise modelling. Indeed, the presented methods seem to miss the effect of the coupling between initial state uncertainty and thruster noise, resulting in an underestimation of the covariance when compared to the results of the MC analysis. In support of this hypothesis, test results showed that the final covariances without the inflation terms are close to the ones obtained only by considering the initial state uncertainty of the Monte Carlo analysis. On the other hand, by considering these inflation terms, the final error is partially mitigated, which is particularly evident in the estimation of the position covariance. Moreover, as highlighted in the introductory section, the mono-dimensional comparison metrics selected (i.e., MISE and TVD) may be penalizing these dynamics-based methods that retrieve a full PDF description even considering coupling terms, whose contribution is completely discarded by these criteria.

Despite these limitations, the results clearly show that the hyperparameters of both methods can be fixed for similar scenarios (i.e., LEO to LEO and LEO DISPOSAL). However, they may vary even for the same orbital regime if the propagation is particularly different, as in the case of SK. In addition, these generalizations seem to be applicable for various orbital regimes: LEO and GEO cases clearly display different ranges of hyperparameters for the A-DAGMMo method. When applied to PoC computation, both methods proved to be effective in capturing the conjunction scenario and the results were always consistent to the reference PoC computed with linearly propagated uncertainties.

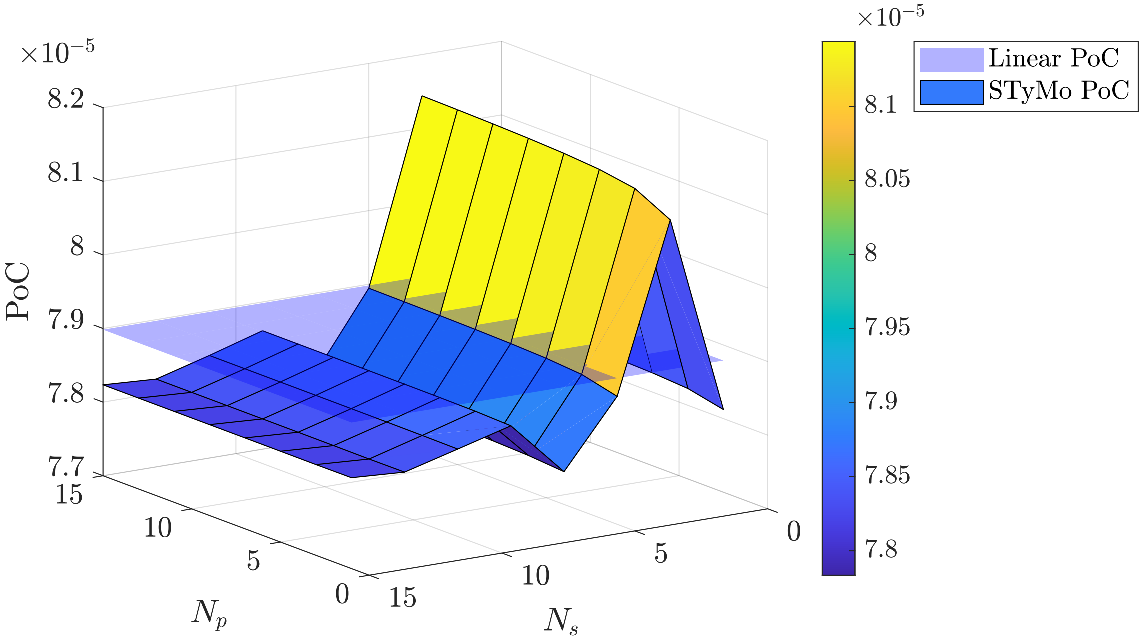


Figure 10. PoC for varying number of GMEs of primary (i.e., N_p) and secondary (i.e., N_s) objects in LEO to LEO scenario.

ACKNOWLEDGMENT

This activity has been carried out under the ESA funded project ELECTROCAM. The authors would like to acknowledge the contribution of the other members of the consortium to this work, starting with GMV for providing the test cases through the work of Pau Gago Padreny and Adrián Díez Martín. Moreover, the authors would like to thank Universidad Carlos III de Madrid for the contribution in modelling the thruster uncertainty provided by Prof. Manuel Sanjurjo Rivo and Prof. Joaquín Míguez.

REFERENCES

[1] K. Holste, P. Dietz, S. Scharmann, K. Keil, T. Henning, D. Zschätzsch, M. Reitemeyer, B. Nauschütt, F. Kiefer, F. Kunze, J. Zorn, C. Heiliger, N. Joshi, U. Probst, R. Thüringer, C. Volkmar, D. Packan, S. Peterschmitt, K. T. Brinkmann, H.-G. Zaunick, M. H. Thoma, M. Kretschmer, H. J. Leiter, S. Schippers, K. Hannemann, and P. J. Klar, "Ion thrusters for electric propulsion: Scientific issues developing a niche technology into a game changer," *Review of Scientific Instruments*, Vol. 91, June 2020, p. 061101, 10.1063/5.0010134.

[2] B. Larbi, M. K. Grzesik, B. Radtke, C. J. Trentlage, and E. Stoll, "Active debris removal for mega constellations: Cubesat possible," *Proceedings of the 9th International Workshop on Satellite Constellations and Formation Flying, IWSCFF2017, 19-21 June, 2017, Boulder, Colorado*, IAF, 2017, pp. 1–19. Paper No. IWSCFF 17-20.

[3] J. Hernando-Ayuso and C. Bombardelli, "Low-Thrust Collision Avoidance in Circular Orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 44, May 2021, pp. 983–995, 10.2514/1.G005547.

[4] J. Gonzalo, C. Colombo, and P. Di Lizia, "A semi-analytical approach to low-thrust collision avoidance manoeuvre design," *Proceedings of the 70th IAC*, Vol. October, 2019. Paper No. IAC-19-A6.2.3.

[5] A. Petit, E. M. Alessi, and A. Rossi, "Low-Thrust Strategies and Implications in the Perspective of Space Debris Mitigation for Large Constellations," *First International Orbital Debris Conference*, Vol. 2109 of *LPI Contributions*, Dec. 2019, p. 6032.

[6] P. S. Maybeck, "Stochastic models, estimation, and control," *Mathematics in Science and Engineering*, Vol. 141, ch. 4, pp. 133–202, Elsevier, 1979, 10.1016/S0076-5392(08)62169-4.

[7] A. T. Fuller, "Analysis of nonlinear stochastic systems by means of the Fokker–Planck equation," *International Journal of Control*, Vol. 9, June 1969, pp. 603–655, 10.1080/00207176908905786.

[8] Y. Sun and M. Kumar, “Uncertainty propagation in orbital mechanics via tensor decomposition,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 124, Mar. 2016, pp. 269–294, 10.1007/s10569-015-9662-z.

[9] A. D. C. Jesus, M. L. D. O. Souza, and A. Prado, “Statistical analysis of nonimpulsive orbital transfers under thrust errors. I,” *Nonlinear Dynamics and Systems Theory*, Vol. 2, Jan. 2002.

[10] Y.-z. Luo and Z. Yang, “A review of uncertainty propagation in orbital mechanics,” *Progress in Aerospace Sciences*, Vol. 89, Feb. 2017, pp. 23–39, 10.1016/j.paerosci.2016.12.002.

[11] R. H. Battin, *An introduction to the mathematics and methods of astrodynamics*. AIAA education series, New York, N.Y.: American Institute of Aeronautics and Astronautics, 1987.

[12] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA, USA: MIT Press, Apr. 1974.

[13] S. Julier and J. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proceedings of the IEEE*, Vol. 92, Mar. 2004, pp. 401–422, 10.1109/JPROC.2003.823141.

[14] I. Arasaratnam and S. Haykin, “Cubature Kalman Filters,” *IEEE Transactions on Automatic Control*, Vol. 54, June 2009, pp. 1254–1269, 10.1109/TAC.2009.2019800.

[15] B. A. Jones, A. Doostan, and G. H. Born, “Nonlinear Propagation of Orbit Uncertainty Using Non-Intrusive Polynomial Chaos,” *Journal of Guidance, Control, and Dynamics*, Vol. 36, Mar. 2013, pp. 430–444, 10.2514/1.57599.

[16] R. S. Park and D. J. Scheeres, “Nonlinear Mapping of Gaussian Statistics: Theory and Applications to Spacecraft Trajectory Design,” *Journal of Guidance, Control, and Dynamics*, Vol. 29, Nov. 2006, pp. 1367–1375, 10.2514/1.20177.

[17] K. Fujimoto, D. J. Scheeres, and K. T. Alfriend, “Analytical Nonlinear Propagation of Uncertainty in the Two-Body Problem,” *Journal of Guidance, Control, and Dynamics*, Vol. 35, Mar. 2012, pp. 497–509, 10.2514/1.54385.

[18] M. Berz, “Differential Algebraic Techniques,” *Advances in Imaging and Electron Physics*, Vol. 108, pp. 81–117, Elsevier, 1999, 10.1016/S1076-5670(08)70228-3.

[19] M. Valli, R. Armellin, P. Di Lizia, and M. R. Lavagna, “Nonlinear Mapping of Uncertainties in Celestial Mechanics,” *Journal of Guidance, Control, and Dynamics*, Vol. 36, Jan. 2013, pp. 48–63, 10.2514/1.58068.

[20] O. Montenbruck and E. Gill, “Satellite Orbits: Models, Methods, and Applications,” *Applied Mechanics Reviews*, Vol. 55, Apr. 2002, pp. B27–B28, 10.1115/1.1451162.

[21] G. Terejanu, P. Singla, T. Singh, and P. D. Scott, “Uncertainty Propagation for Nonlinear Dynamic Systems Using Gaussian Mixture Models,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, Nov. 2008, pp. 1623–1633, 10.2514/1.36247.

[22] 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*, Vol. 36, July 2013, pp. 1047–1057, 10.2514/1.58987.

[23] V. Vittaldev and R. P. Russell, “Space Object Collision Probability Using Multidirectional Gaussian Mixture Models,” *Journal of Guidance, Control, and Dynamics*, Vol. 39, Sept. 2016, pp. 2163–2169, 10.2514/1.G001610.

[24] Z.-J. Sun, Y.-Z. Luo, P. Di Lizia, and F. B. Zazzera, “Nonlinear orbital uncertainty propagation with differential algebra and Gaussian mixture model,” *Science China Physics, Mechanics, and Astronomy*, Vol. 62, Mar. 2019, p. 34511, 10.1007/s11433-018-9267-6.

[25] A. Fossà, R. Armellin, E. Delande, M. Losacco, and F. Sanfedino, “Multifidelity Orbit Uncertainty Propagation using Taylor Polynomials,” *AIAA SCITECH 2022 Forum*, San Diego, CA & Virtual, American Institute of Aeronautics and Astronautics, Jan. 2022, 10.2514/6.2022-0859.

[26] M. Rasotto, A. Morselli, A. Wittig, M. Massari, P. D. Lizia, R. Armellin, C. Valles, and G. Ortega, “Differential algebra space toolbox for nonlinear uncertainty propagation in space dynamics,” *The 6th International Conference on Astrodynamics Tools and Techniques (ICATT) (14/03/16 - 17/03/16)*, March 2016.

[27] L. Pardo, *Statistical inference based on divergence measures*, ch. 1, p. 51. Chapman and Hall/CRC, 2018.

[28] J. L. Junkins, M. R. Akella, and K. T. Alfriend, “Non-Gaussian error propagation in orbital mechanics,” *Journal of The Astronautical Sciences*, Vol. 44, 1996, pp. 541–563.

[29] 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*, Vol. 122, Jul 2015, pp. 239–261, 10.1007/s10569-015-9618-3.

[30] L. G. Jacchia, “Thermospheric Temperature, Density, and Composition: New Models,” *SAO Special Report*, Vol. 375, Mar. 1977.

[31] J. M. Picone, A. E. Hedin, D. P. Drob, and A. C. Aikin, “NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues,” *Journal of Geophysical Research: Space Physics*, Vol. 107, No. A12, 2002, pp. SIA 15–1–SIA 15–16, https://doi.org/10.1029/2002JA009430.

[32] K. Chan, “Improved analytical expressions for computing spacecraft collision probabilities,” *Advances in the Astronautical Sciences*, Vol. 114, 2003, pp. 1197–1216.