A. Morselli, R. Armellin, P. Di Lizia, F. Bernelli Zazzera
A High Order Method for Orbital Conjunctions Analysis: Monte Carlo Collision Probability Computation
Advances in Space Research, Vol. 55, N. 1, 2015, p. 311-333
doi:10.1016/j.asr.2014.09.003

A high order method for orbital conjunctions analysis: Monte Carlo collision probability computation

Alessandro Morselli[1,*]

Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy

Roberto Armellin[2]

Aeronautics, Astronautics and Computational Engineering Unit, University of Southampton, Highfield campus, SO17 1BJ, Southampton, United Kingdom

Pierluigi Di Lizia[3], Franco Bernelli Zazzera[4]

Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Via La Masa 34, 20156 Milano, Italy

Abstract

Three methods for the computation of the probability of collision between two space objects are presented. These methods are based on the high order Taylor expansion of the time of closest approach (TCA) and distance of closest approach (DCA) of the two orbiting objects with respect to their initial conditions. The identification of close approaches is first addressed using the nominal objects states. When a close approach is identified, the dependence of the TCA and DCA on the uncertainties in the initial states is efficiently computed with differential algebra (DA) techniques. In the first method the collision probability is estimated via fast DA-based Monte Carlo simulation,

*Corresponding author (tel: +39-02-2399-8401 , fax: +39-02-2399-8334)

Email addresses: alessandro.morselli@polimi.it (Alessandro Morselli), roberto.armellin@soton.ac.uk (Roberto Armellin), pierluigi.dilizia@polimi.it (Pierluigi Di Lizia), franco.bernelli@polimi.it (Franco Bernelli Zazzera)

[1]PhD candidate

[2]Lecturer

[3]Postdoc fellow

[4]Full Professor

in which, for each pair of virtual objects, the DCA is obtained via the fast evaluation of its Taylor expansion. The second and the third methods are the DA version of Line Sampling and Subset Simulation algorithms, respectively. These are introduced to further improve the efficiency and accuracy of Monte Carlo collision probability computation, in particular for cases of very low collision probabilities. The performances of the methods are assessed on orbital conjunctions occurring in different orbital regimes and dynamical models. The probabilities obtained and the associated computational times are compared against standard (i.e. not DA-based) version of the algorithms and analytical methods. The dependence of the collision probability on the initial orbital state covariance is investigated as well.

Keywords: Space Debris; Orbital conjunction; Collision probability; Differential Algebra

---

1. Introduction

The risk of in-orbit collisions between operative satellites and space debris is a crucial issue in satellite operation. When a close approach is identified, it is necessary to define an indicator that can tell how risky the predicted conjunction is. It is common practice for space agencies and satellite operators to consider, together with conjunction geometry and miss-distance, the collision probability for this purpose (Klinkrad et al., 2005; Righetti et al., 2011).

The collision probability is computed by means of a multi-variate integral. *** The uncertainties in position and velocity coming from orbit determination can be translated into a probability density function (p.d.f.). *** The probability density function is then integrated over the volume swept out by the combined hard-body area of the satellite and colliding object, normal to the velocity vector, to retrieve the collision probability.

Different methods exist for the computation of this multi-dimensional integral. Most of these approaches (Akella and Alfriend, 2000; Bèrend, 1999; Patera, 2001; Klinkrad, 2006) have the following assumptions in common:

- Position uncertainties of the two objects are not correlated;
- Objects move along straight lines at constant velocity during the conjunction;

- The uncertainty in the velocities is neglected;
- Position uncertainty during the whole encounter is constant and equal to the value *** during the conjunction; ***
- The uncertainties in the positions of the two objects are represented by three-dimensional Gaussian distributions.

These assumptions produce accurate results when the relative motion between the satellite and the object is *** rectilinear and the conjunction occurs close to the initial epoch so that the p.d.f. of the relative position of the two objects remains Gaussian. *** The probability density function in the proximity of the close approach, under the assumption that position error is Gaussian, is expressed as

$$p\left(\Delta \boldsymbol{r}\right) = \frac{1}{\sqrt{\left(2\pi\right)^3 \det \boldsymbol{C}}} e^{-\frac{1}{2}\Delta \boldsymbol{r}^T \boldsymbol{C}^{-1} \Delta \boldsymbol{r}}, \tag{1}$$

where $\Delta \boldsymbol{r}$ is the objects relative position vector. Integrating over the volume V swept out by the *** hard-body sphere with volume V_c ***, that is the combined volume of the colliding objects, yields the collision probability

$$P_c = \frac{1}{\sqrt{\left(2\pi\right)^3 \det \boldsymbol{C}}} \iiint_V e^{-\frac{1}{2}\Delta \boldsymbol{r}^T \boldsymbol{C}^{-1} \Delta \boldsymbol{r}} \, dV. \tag{2}$$

*** Because of the assumption of rectilinear motion of both conjuncting objects ***, the volume V is a cylinder extending along the relative velocity direction. By integrating the p.d.f. along the cylinder axis from -∞ to +∞, the marginal two-dimensional p.d.f is obtained and the volume integral is reduced to a two-dimensional integral on the collision cross sectional area (Chan, 2008). Supposing that the combined covariance *** $\boldsymbol{C}$ *** is centered on the primary object and that the combined hard-body is positioned on the *** secondary object ***, the two-dimensional integral of the marginal p.d.f. on the collision *** cross-sectional area in the (x,y) encounter plane can be written as (Akella and Alfriend, 2000; Klinkrad, 2006; Bèrend, 1999): ***

$$P_c = \frac{1}{2\pi\sqrt{\det \boldsymbol{C}}} \int_{-R_c}^{R_c} \int_{-\sqrt{R_c^2 - x^2}}^{\sqrt{R_c^2 - x^2}} e^{-A} \, dy \, dx, \tag{3}$$

where

$$A = \frac{1}{2} \Delta \boldsymbol{r}^T \, \boldsymbol{C}^{-1} \Delta \boldsymbol{r}, \tag{4}$$

where R_c is the combined radius of the two spherical objects and C now denotes the covariance in the marginal two-dimensional pdf.*** The analytical methods available in the literature differ in the way the two-dimensional integral is approximated. Chan transforms the two-dimensional p.d.f. into a one-dimensional Rician p.d.f. and uses equivalent areas to develop an analytical approximation of the double integral (Chan, 1997). A series expression to approximate Eq. (3) is derived by Alfano, using a combination of error functions and exponential terms (Alfano, 2006b). In addition, Patera performs an exact reduction of the two-dimensional integral of Eq. (3) to a one-dimensional contour integral over a general-shaped body (Patera, 2001). The method was then extended to use numerical quadrature for a simple one-dimensional integral (Patera, 2005).

Methods that account for non-linearities, which are typical of GEO conjunctions, were also developed (Chan, 2004; Patera, 2003, 2006). An approach that uses a set of small consecutive linear segments to compute collision probability for non-linear conjunctions is presented in (Alfano, 2006a; McKinley, 2006).

The conflict probability, used for air-traffic control by the aviation community (Paielli and Erzberger, 1997), was proposed as an alternative to collision probability as a metric to quantify the collision risk even for space objects (Patera, 2007a). The conflict probability is computed similarly to collision probability, using a conflict volume instead of the combined hard-body region. It corresponds to the probability that a single conflict volume, centered on one space object, will be penetrated by the other space object. The conflict volume is large compared to space vehicle size and, as a result, conflict probability is higher than collision probability. In addition, no information on hard-body size, which is usually not available for space debris, is required. The conflict probability was extended to the case of ellipsoidal conflict volumes and tested against other metrics for the identification of risky conjunctions, showing good performances for the analyzed test cases (Patera, 2007b).

Besides the analytical methods, the collision probability integral can be computed by means of Monte Carlo (MC) simulations (de Vries and Phillion, 2010; Sabol et al., 2011). Despite being a general and flexible way to compute collision probability, the MC approach has the main drawback of requiring intensive computation, as each virtual satellite/debris trajectory has to be propagated. For this reason Monte Carlo methods are not suitable for daily collision probability computation, since results can be obtained in a

timely manner only with simple dynamics, such as two-body propagators or SGP4/SDP4.

In recent times, techniques such as importance sampling (Dolado et al., 2011) or adaptive splitting (Pastel, 2011) have been introduced to cope with the high computational effort. Moreover, a method that couples Monte Carlo with orbital dynamics approximation, obtained by means of polynomial chaos expansion, was introduced to compute satellite collision probability with reduced computational effort (Jones and Doostan, 2013). Monte Carlo methods were also used to study the impact of non-Gaussian *** probability density functions *** on collision probability computation (Ghrist and Plakalovic, 2012).

New methods to reduce the computational effort related to collision probability computation are presented in this work. These methods are based on the Taylor expansion of the TCA and the Distance of Close Approach (DCA) of the two orbiting objects. The occurrence of close approaches is first identified using the nominal initial orbital states. Then, differential algebra (DA) techniques are used to propagate sets of initial conditions by computing the Taylor approximation of the final states at the nominal TCA. The polynomial expansion of the TCA with respect to uncertainties in the initial states is obtained by means of polynomial inversion tools and plugged into the DCA and final state maps to retrieve their dependence on initial uncertainties (Morselli et al., 2014). The methods for collision probability computation can now take advantage of the availability of the resulting polynomial maps. More in detail, the initial positions and velocities are sampled according to their estimated uncertainties. For each pair of virtual objects, the associated DCA is computed through the fast evaluation of its Taylor expansion rather than running computationally intensive numerical integrations. The DCA is then compared with the collision threshold, i.e. the diameter of the sphere that envelopes the two objects. Three methods for the computation of collision probability are presented in this paper: a DA-based Monte Carlo simulation and the DA version of two advanced techniques, namely Line Sampling (LS) and Subset Simulation (SS) (Au and Beck, 2001; Koutsourelakis et al., 2004).

The manuscript is organized as follows. First the description of MC, LS, and SS methods in their standard version (i.e. not DA-based) is given in Sections 2.1, 2.2, and 2.3. Then, their formulation taking advantage of DA techniques is introduced in Section 3. Numerical examples and discussion of results are provided in Section 4, which is followed by conclusions.

2. Methods for collision probability computation

The two objects initial conditions are defined by the state vectors $\boldsymbol{x}_0^1$ and $\boldsymbol{x}_0^2$. Both vectors are uncertain due to the orbit determination process, and their statistics can be represented by a pdf. Typically, the initial statistics is assumed to be Gaussian, then the initial state is fully described by its mean and covariance matrix.

The methods for collision probability computation described in this section rely on the Performance Function (PF)

$$g\left(\boldsymbol{x}_0^1, \boldsymbol{x}_0^2\right) = D - d^*\left(\boldsymbol{x}_0^1, \boldsymbol{x}_0^2\right), \tag{5}$$

in which D is the collision threshold, and d^* is the function that maps each pair of initial conditions $\boldsymbol{x}_0^1, \boldsymbol{x}_0^2$ to the associated DCA. Note that, as D is a constant and d^* is the distance between the the objects centers of mass it follows that exact collision probabilities are computed for spherical space objects only. According to the definition of the PF, the following conditions occur

$$g\left(\boldsymbol{x}_0\right) \begin{cases} < 0 & \Rightarrow \quad \text{no collision} \\ = 0 & \Rightarrow \quad \text{at limit state} \\ > 0 & \Rightarrow \quad \text{collision,} \end{cases} \tag{6}$$

where, for the sake of brevity, $\boldsymbol{x}_0 = (\boldsymbol{x}_0^1, \boldsymbol{x}_0^2)$.

The collision threshold D can be related to the dimensions of the two objects. Let L_i, for $i = 1, 2$, be the diameters of the spherical objects. Then, according to Figure 1, D is given by the sum of the radius of the two objects. In case of non-spherical objects without large appendages the same performance function could be used, by selecting the sphere that envelopes each object i. This is a conservative approach, which guarantees that the resulting collision probability is larger than its correct value since the collision condition will hold for a larger number of virtual objects.

2.1. Monte Carlo method

To compute the collision probability by means of MC simulation, the initial position and velocity of the two objects are sampled from their error covariance matrices (thus generating what are called virtual objects or debris). The initial orbital states, $\boldsymbol{x}_0^1$ and $\boldsymbol{x}_0^2$, are then propagated till the time derivative $\dot{d}$ of the relative distance d is zero, which happens in the surrounding of the nominal TCA. In this way, the TCA and DCA are identified for

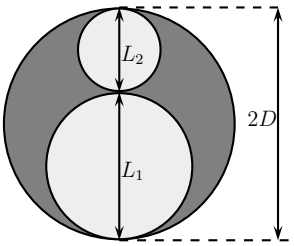


Figure 1: Collision threshold definition

each pair of virtual objects. If the relative distance is below the threshold D a hit is counted. The number of samples N_c for which the collision condition is verified, i.e. $g_x(\boldsymbol{x}_0) > 0$, is divided by the total number of samples N_T to compute the collision probability

$$P(d^* < D) = \frac{N_c}{N_T}. \tag{7}$$

The standard deviation of the computed probability is given by

$$\sigma = \sqrt{\frac{P(1-P)}{N_T}}, \tag{8}$$

and is proportional to $1/\sqrt{N_T}$. For standard Monte Carlo methods the coefficient of variation (c.o.v.), i.e. the ratio between the standard deviation and the mean value, is thus defined as

$$\delta(P) = \frac{\sigma}{P} = \sqrt{\frac{1-P}{N_T P}}. \tag{9}$$

The collision probability between two spacecraft is usually very low since it exceeds 10^{-4} only for really close conjunctions. As a consequence, a large number of samples is required to obtain a sufficiently accurate estimate of its value. According to Dagum et al. (2000), the number of samples N_T to be used in a Monte Carlo simulation when $\sigma^2 > \varepsilon P_c$ should be at least

$$N_T > \frac{4(e-2)(1-P_c)}{P_c \varepsilon^2} \log\left(\frac{2}{\beta}\right), \tag{10}$$

where P_c is the collision probability, $(1-\beta)$ is the desired confidence level, and ε is the relative error of the collision probability. The number of samples

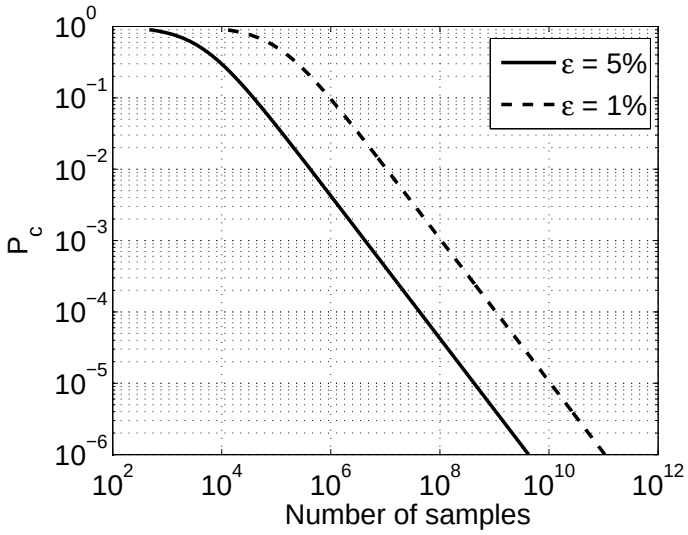


Figure 2: Number of Monte Carlo samples required to compute a probability P_c with a 95% confidence level and relative error ε

required to compute a given collision probability with a 95% confidence level is illustrated in Figure 2, where the dashed and a solid lines are computed with a relative error of 1% and 5%, respectively.

To compute a collision probability of 10^{-4} with a relative error of 5% at least 4.24×10^7 samples are required, whereas for $P_c = 10^{-6}$ the minimum number of samples increases to 4.24×10^9.

These considerations point out the main drawback of MC simulations, that is the high computational effort which is magnified when dealing with very low probabilities or when a computationally intensive simulation is required for each sample, such as a numerical integration of the equation of motion. Line Sampling and Subset Simulation algorithms, described in the next two subsections, were developed to reduce the number of samples required to compute low collision probabilities.

2.2. Line Sampling

The main idea behind LS is transforming a high dimensional problem into a number of conditional one-dimensional problems solved along an "important direction" $\boldsymbol{\alpha}$ (Koutsourelakis et al., 2004). The key issue of the method is identifying this direction, that should point toward the region of failure, i.e. the hyper-volume of position and velocity deviations for which the collision

criterion holds. The important direction tells which combination of states variations is more efficient to reach the failure condition.

In the LS approach, the vector of uncertain parameters $\boldsymbol{x}_0 \in \mathbb{R}^n$, where n is the number of uncertain parameters, has first to be transformed into the adjoint vector $\boldsymbol{\theta} \in \mathbb{R}^n$. This vector belongs to the so-called "standard normal space", where each variable is represented by an independent central unit Gaussian distribution. This is done using Rosenblatt's transformation (Rosenblatt, 1952)

$$\begin{aligned} \boldsymbol{\theta} &= T_{x,\theta}\left(\boldsymbol{x}_0\right) \\ \boldsymbol{x}_0 &= T_{\theta,x}\left(\boldsymbol{\theta}\right), \end{aligned} \tag{11}$$

where the operator $T_{\cdot,\cdot}$ indicates the transformation, and applying it to the performance function

$$g\left(\boldsymbol{x}_0\right) = g_x\left(T_{\theta,x}\left(\boldsymbol{\theta}\right)\right) = g_\theta\left(\boldsymbol{\theta}\right). \tag{12}$$

A natural choice for the important direction is the normalized gradient of the PF at the nominal point in the standard normal space

$$\boldsymbol{\alpha} = \frac{\boldsymbol{\nabla}_{\boldsymbol{\theta}}\, g_\theta\left(\boldsymbol{\theta}\right)}{\left\|\boldsymbol{\nabla}_{\boldsymbol{\theta}}\, g_\theta\left(\boldsymbol{\theta}\right)\right\|_2}. \tag{13}$$

If not available analytically, this gradient can be numerically estimated. The more the estimate of the important direction is close to its true value, the lower will be the variance of the failure probability (Pradlwarter et al., 2005). For high-dimensional problems where the numerical computation of gradients can be time-demanding, it is possible to obtain an estimate by computing the normalized "center of mass" of the failure domain. This is achieved by Monte Carlo Markov Chain (MCMC), using as seed a point belonging to the failure region or close to it and computing the mean of the N_α samples generated in the failure region (Zio and Pedroni, 2009b).

Once the important direction is identified, the LS method proceeds as follows

LS 1. Sample N_T vectors $\boldsymbol{\theta}^i$ from the normal multidimensional joint probability distribution.

LS 2. Estimate for each sample its conditional one-dimensional failure probability $\hat{P}^{1D,i}$ performing the following operations

(a) Project the vector $\boldsymbol{\theta}^i$ onto the straight line passing through the origin and perpendicular to $\boldsymbol{\alpha}$ to obtain vector $\boldsymbol{\theta}^{i,\perp}$, as portrayed in Figure 3(a).

(b) Write the parametric equation of samples along the important direction, $\tilde{\boldsymbol{\theta}}^i = \boldsymbol{\theta}^{i,\perp} + c\,\boldsymbol{\alpha}$, as sketched in Figure 3(b).

(c) Compute the values of c_j^i, $j = 1, 2$, for which the PF is equal to zero. (Here and in the remainder of the paper a maximum number of two zeros is considered.) This step requires evaluations of the PF, which involve numerical propagations or complex system simulations when g_θ is not known analytically. Note that when the failure region is infinite only one real solution is found whereas the other is $+\infty$.

(d) If the two values coincide or no solution is found then the i-th one-dimensional probability $\hat{P}^{1D,i}$ is equal to zero; else, given the two solutions c_1^i and c_2^i, with $c_1^i > c_2^i$, the probability is

$$\begin{aligned} \hat{P}^{1D,i}(F) &= P\left[c_2^i \leq N(0,1) \leq c_1^i\right] = \\ &= \Phi\left(c_1^i\right) - \Phi\left(c_2^i\right) \end{aligned} \tag{14}$$

where $\Phi\left(c_j^i\right)$ is the standard normal cumulative distribution function, $N(0,1)$ is the standard normal distribution, with zero mean and unit standard deviation, and F indicates the collision condition $d \leq D$.

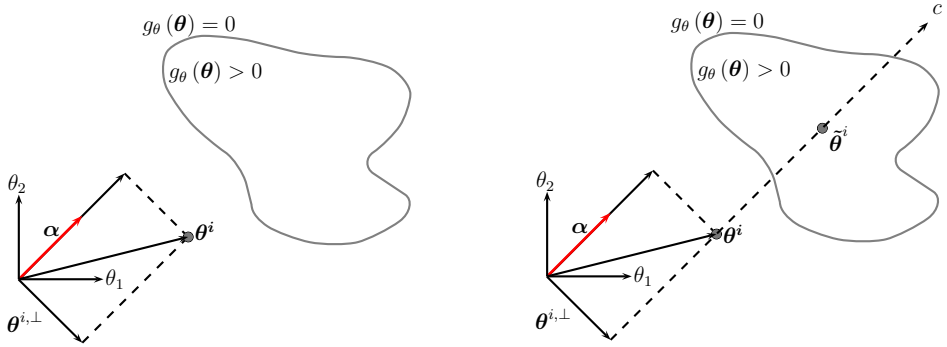
LS 3. Compute the unbiased estimator $\hat{P}^{N_T}(F)$, which is the sample average of the independent conditional one-dimensional probability estimate

$$\hat{P}^{N_T}(F) = \frac{1}{N_T} \sum_{i=1}^{N_T} \hat{P}^{1D,i}(F) \tag{15}$$

The variance of the collision probability in Eq. (15) is given by

$$\sigma^2\left(\hat{P}^{N_T}(F)\right) = \frac{1}{N_T\left(N_T - 1\right)} \sum_{i=1}^{N_T} \left(\hat{P}^{1D,i}(F) - \hat{P}^{N_T}(F)\right)^2 \tag{16}$$

The total number of system simulations is related to the number of PF evaluations required to compute, for each sample $\boldsymbol{\theta}^i$, the values c_j^i at step



(a) Projection of the sample vector $\boldsymbol{\theta}^i$ on directions parallel and perpendicular to $\boldsymbol{\alpha}$ to obtain $\boldsymbol{\theta}^{i,\perp}$

(b) Graphical representation of the line with parametric equation $\tilde{\boldsymbol{\theta}}^i(c)$ used to identify the values of c for which $g_\theta(c) = 0$

Figure 3: Illustration of the Line Sampling procedure in a bi-dimensional space (θ_1, θ_2). The origin of the reference frame is in the nominal initial states and $\boldsymbol{\alpha}$ points in the direction of the gradient (i.e. greatest rate of decrease) of the relative distance. Failure region is surrounded by the grey line.

LS 2(c). When the PF is smooth and does not present oscillations along the important direction, the number of sample evaluations can be limited to the one necessary to obtain an approximation of the function in the region of interest (Zio and Pedroni, 2009b). For a short-term encounter between two space objects, the PF along the direction $\boldsymbol{\alpha}$ resembles a parabola. With three evaluations of the PF, it is indeed possible to obtain a second-order approximation and compute an approximate value of the intersections c_1^i and c_2^i with the line $g_\theta(c) = 0$, if they exists. Although reducing the computational effort when dealing with computationally demanding simulations, such an approach strongly depends on the choice of the c grid required to compute the approximation of the PF. A wrong spacing of the grid could result in erroneous estimations of the intersections with the failure region boundary. Thus, when this approach is selected, it is important to verify that the computed intersections are close to the true values for different close encounter geometries and relative velocities.

2.3. Subset Simulation

Subset Simulation (SS) is an adaptive stochastic simulation method to compute efficiently small failure probabilities (Au and Beck, 2001). The idea at the basis of the method is to compute the probability as a product

of larger conditional probabilities. Thus, given a sequence of intermediate failure regions $F_1 \supset F_2 \supset \cdots \supset F_m = F$, the failure probability becomes

$$P(F) = P(F_m) = P(F_1) \prod_{l=1}^{m-1} P\left(F_{l+1}|F_l\right), \tag{17}$$

where $P(F_{l+1}|F_l)$ indicates the probability of F_{l+1} conditional to F_l. The method is initialized using a standard Monte Carlo simulation to generate samples at conditional level 0. Once the failure region F_1 is determined and the probability $P(F_1)$ computed, a Monte Carlo Markov Chain (MCMC) algorithm (Metropolis et al., 1953) is used to generate samples conditional to the failure region F_1. Another intermediate failure region F_2 is then located and other samples are generated with MCMC. The process can be repeated till the failure region corresponding to objects collision is identified. The approach was originally developed to address structural failure, but it was also used in different research areas in reliability, e.g. to address the failure probability of thermal-hydraulic passive system (Zio and Pedroni, 2009a).

The main issue of the algorithm is to identify the intermediate failure regions. This can be achieved by choosing a constant probability value p_0, and searching a threshold value of the relative distance at every conditional level for which $P(F_l|F_{l-1}) = p_0$. The PF changes accordingly: being D_l the collision threshold at conditional level l, it can be defined as

$$g_x^l\left(\boldsymbol{x}_0\right) = D_l - d^*\left(\boldsymbol{x}_0\right). \tag{18}$$

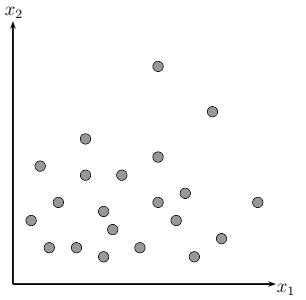
Similarly to Eq. (5), the following conditions occur

$$g_x^l\left(\boldsymbol{x}_0\right) \begin{cases} < 0 & \Rightarrow \quad \boldsymbol{x}_0 \text{ is out of } l\text{-th conditional level} \\ = 0 & \Rightarrow \quad \boldsymbol{x}_0 \text{ is at limit state} \\ > 0 & \Rightarrow \quad \boldsymbol{x}_0 \text{ is in } l\text{-th conditional level.} \end{cases} \tag{19}$$

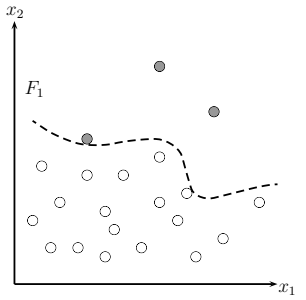
Since the conditional probability is equal to p_0 at each iteration, the collision probability in Eq. (17) can be computed as

$$P(F) = P(F_m) = P\left(F_m|F_{m-1}\right) \; {p_0}^{m-1} \tag{20}$$

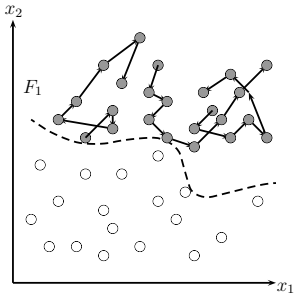
The resulting SS algorithm goes through the following steps (refer to Figure 4 for its schematic representation):



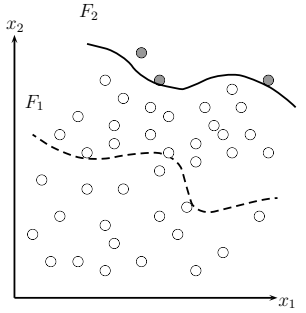
(a) Conditional level 0: Monte Carlo simulation.



(b) Adaptive selection of samples in conditional level 1.



(c) MCMC simulation: generation of new samples belonging to conditional level 1.



(d) Adaptive selection of samples in conditional level 2.

Figure 4: Illustration of the Subset Simulation algorithm for a bi-dimensional space (x_1, x_2). The dots are the generated samples and the lines identify the conditional levels. The grey dots are the samples belonging to the l-th conditional levels and the arrows represent the MCMC path.

SS 1. Set $l = 0$ and generate N sample vectors $\boldsymbol{x}_0^{0,k}$, $k = 1, \ldots, N$, by standard MC simulation. The superscript 0 denotes that the samples are at "conditional level 0".

SS 2. Compute the values of the PF $g_x^l(\boldsymbol{x}_0)$ for the N samples $\boldsymbol{x}_0^{l,k}$.

SS 3. Sort the N samples in ascending order, according to their associated value of the performance function g_x^l. The samples closer to the failure region will be at the bottom of the list.

SS 4. Choose the intermediate threshold value D_{l+1} from the $(1 - p_0)N$-th value of the sorted list. The $(l + 1)$-th conditional level is then defined as $F_{l+1} = \{d^* < D_{l+1}\}$. By definition the associated conditional

probability is $P\left(F_{l+1}|F_l\right) = P\left(d^* < D_{l+1} \,|\, d^* < D_l\right) = p_0$.

SS 5. If $D_{l+1} \leq D$ go to last step otherwise identify the $p_0 N$ samples $\boldsymbol{x}_0^{l,u}$, $u = 1, 2, \ldots, p_0 N$, whose relative distance lies in F_{l+1}. All these samples belong to "conditional level $l+1$".

SS 6. Using MCMC, generate $(1-p_0)N$ additional conditional samples distributed as $p\left(\cdot|F_{l+1}\right)$, so that a total of N conditional samples $\boldsymbol{x}_0^{l+1,k} \in F_{l+1}$, where $k = 1, \ldots, N$. Eq. (19) can be used to establish whether each sample belongs to conditional level $l+1$ or not.

SS 7. Set $l = l+1$ and return to step 2 above

SS 8. Stop the algorithm

The total number of samples generated is

$$N_T = N + (m-1)(1-p_0)N, \tag{21}$$

where m is the number of conditional levels required to reach the failure region. According to Eq. (20) the collision probability becomes

$$P\left(d^* < D\right) = {p_0}^{m-1} P\left(F_m \,|\, F_{m-1}\right) = {p_0}^{m-1} \frac{N_F}{N}, \tag{22}$$

where N is the total number of samples at each conditional level and N_F is the number of samples at conditional level m, whose relative distance is less than the collision threshold D.

The efficiency of the SS algorithm relies on the proper selection of its parameters: the conditional failure probability p_0, the number of samples of each step N, and the shape of the proposal probability density function for the generation of the Markov chain. A detailed analysis on the selection of these parameters is given in (Zuev et al., 2012), where it is shown that the optimal choice for p_0 is 0.2 (although similar efficiency is obtained for $p_0 \in [0.1; 0.3]$). For what concerns the proposal p.d.f. of MCMC, a univariate Gaussian distribution is used in this work. The variance σ_l of the proposal p.d.f. is changed dynamically at each conditional level l so that the acceptance rate of Markov Chain samples is kept between 30% and 50%. This solution is nearly optimal, i.e. the chain converges to stationarity nearly as fast as possible.

Zuev et al. (2012) also suggest a Bayesian post-processor for Subset Simulation, SS+, to refine the computed failure probability and determine higher moments, allowing the computation of the failure probability variance. Defining

$$n_l = \begin{cases} p_0 N & \text{if } l < m \\ N_F & \text{if } l = m \end{cases} \tag{23}$$

the first moment of the distribution of the failure probability becomes

$$E_{SS+}\left[P(F)\right] = \prod_{l=1}^{m} \frac{n_l + 1}{N + 2}, \tag{24}$$

whereas the second moment is given by

$$E_{SS+}\left[P(F)^2\right] = \prod_{l=1}^{m} \frac{(n_l + 1)\,(n_l + 2)}{(N + 2)\,(N + 3)}. \tag{25}$$

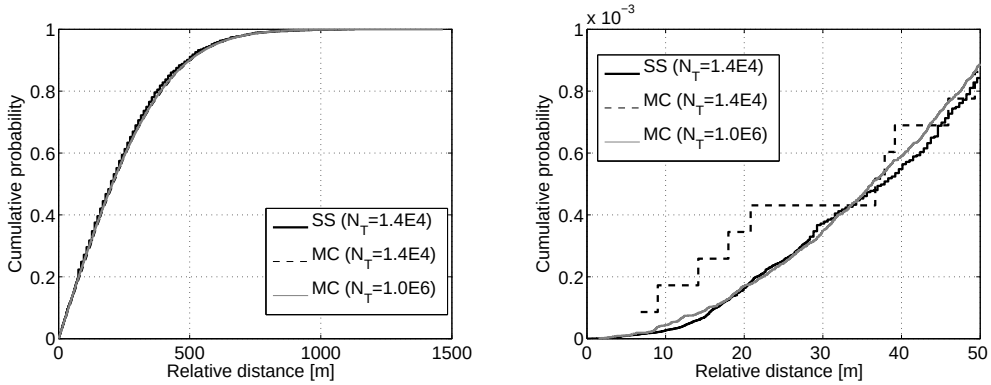
The variance of the collision probability $P(F)$ can then be obtained using the definition

$$Var(P) = E[P^2] - \left(E[P]\right)^2. \tag{26}$$

The number of samples N to be used depends on the problem dimension and the expected failure probability. Furthermore, if the failure region is disconnected, the samples must be dense enough to lie in the proximity of each subregion at conditional level zero and then reach them at the subsequent conditional levels. The tests showed that a good choice is $N = 2 \times 10^3$ for collision probability computation.

To stress the advantages of the SS method over standard MC for low failure probability computation, two experimental cumulative distribution functions (c.d.f.) obtained with SS and MC for a close conjunction are compared in Figure 5. The solid black line is the distribution obtained with SS using 14000 samples ($p_0 = 0.2$, $N = 2000$ samples and 7 conditional levels), the dashed line is the c.d.f. obtained with MC using the same number of samples, and the solid grey line is the c.d.f. for a MC with 10^6 samples.

The three curves seem to agree over the entire set of relative distances. However, the detail reported in Fig. 5(b) shows the lack of accuracy of the MC simulation with fewer samples in the proximity of the failure region. As the maximum cumulative probability in Fig. 5(b) is about 0.001, only 14 samples out of 14000 lie, on average, in the associated range of relative



(a) Experimental c.d.f.

(b) Detail of experimental c.d.f. near failure region

Figure 5: Cumulative probability comparison between SS and MC

distances in the MC simulation. In particular, only two samples have a relative distance below 10 m and no samples are located under 5 m. In contrast, SS generates nearly 5000 samples in the same region, which is five times more than the ones generated by MC with 10^6 samples. The samples generated by SS provide enough information to describe the c.d.f. accurately even at lower conditional probability level, i.e. closer to the failure region.

3. DA-based methods for collision probability

The methods described in the previous section are here modified to take advantage of the Taylor expansion of the DCA. The computation of the polynomial approximation is achieved by means of the DA techniques implemented in COSY INFINITY. Differential algebra supplies the tools to compute the derivatives of functions within a computer environment. More specifically, by substituting the classical implementation of real algebra with the implementation of a new algebra of Taylor polynomials, any multivariate function is expanded into its Taylor series up to an arbitrarily order with limited computational effort. As a main consequence, the Taylor expansion of the solution of any ordinary differential equation can be obtained by carrying out all the operations of any explicit integration scheme in the DA framework. Thus, the dependence of the solution of the ordinary differential equation on initial conditions and time is available in terms of high order polynomial maps. For details on the DA mathematical foundation and on

its practical use the reader can refer to Berz (1999) and Berz and Makino (2011). In the following, the key points for the computation of the high order Taylor series expansion of the DCA and TCA are first given. Then, Section 3.2 illustrates the DA implementation of the methods for collision probability.

3.1. High order expansion of DCA and TCA

A DA algorithm has been developed by the authors to compute the arbitrary order Taylor expansions of TCA and DCA with respect to uncertain initial conditions in a general dynamical model (Armellin et al., 2012). As presented in detail in Morselli et al. (2014), the Taylor expansions of the state vectors $\boldsymbol{x}_f^1$ and $\boldsymbol{x}_f^2$ of the two objects at the nominal TCA are first obtained by propagating their initial conditions with a DA-based numerical integrator. The resulting polynomials are functions of both the final time and the initial uncertain state vectors $\boldsymbol{x}_0^1$ and $\boldsymbol{x}_0^2$. The polynomial map of the relative distance between the two objects is then computed through simple algebraic manipulations. By using partial polynomial inversion techniques and imposing the stationarity condition of the relative distance with respect to time, the Taylor expansion of TCA and DCA with respect to $\boldsymbol{x}_0^1$ and $\boldsymbol{x}_0^2$,

$$[t^*] = t^* + \mathcal{M}_{t^*}\left(\delta\boldsymbol{x}_0^1, \delta\boldsymbol{x}_0^2\right) \tag{27}$$

$$[d^*] = d^* + \mathcal{M}_{d^*}\left(\delta\boldsymbol{x}_0^1, \delta\boldsymbol{x}_0^2\right), \tag{28}$$

are computed. For any perturbed initial condition of the two objects (i.e., for any pair of virtual debris), the evaluation of the Taylor polynomials in (27) and (28) delivers the associated values of TCA and DCA. Consequently, the main idea is to reduce the computational cost of standard methods for the computation of collision probability by replacing multiple – computationally intensive – numerical integrations with multiple – fast – evaluations of the polynomial maps (27) and (28).

As a last remark note that two main factors affect the accuracy of the results: the dynamical model used for the propagation of the states and the accuracy of the Taylor expansion. For the first issue the DA-based numerical propagator AIDA developed in Morselli et al. (2014) is used in this work. The implemented dynamical model includes

- the gravitational model EGM2008 up to order 10,
- the atmosphere model NRLMSISE-00 to compute air density,

- third body perturbations,
- and solar radiation pressure with a dual-cone model for Earth shadow.

The accuracy of the polynomial maps (27) and (28) depends on the expansion order, the initial uncertainties, and the propagation window. In Morselli et al. (2014) the accuracy of the maps was assessed by considering the 100 samples with the largest displacement from the nominal initial conditions among a larger set of 10^9 samples generated considering the full initial covariance matrix. It was shown that, for typical values of initial uncertainties and for a maximum propagation window of one week, the error of the Taylor expansion of the DCA is less than 1 m for a third order expansion for both LEO and GEO orbital regimes.

3.2. DA-based methods

Once the 12-variables kth order Taylor expansions of the TCA and DCA are available, the methods for collision probability computation can be easily modified to work on the resulting polynomials. It is worth observing that the presented methods are applicable only for cases with a single DCA and TCA. Nonlinear relative motion with multiple DCAs are not analyzed in this paper and will be addressed in future works. In the following, the main modifications to the three proposed methods are summarized.

DAMC. For what concerns the Monte Carlo method, its DA-based counterpart is simply obtained by substituting each pair of numerical or analytical propagations necessary to compute the DCA with a single evaluation of the map in Eq. (28). In the following, the acronym DAMC-k is used to label the resulting DA-based Monte Carlo method, where k is the order used for the Taylor expansion of the TCA and DCA.

DALS. The availability of the DCA expansion is exploited for the computation of the important direction $\boldsymbol{\alpha}$ in the LS method. Once the polynomial map in Eq. (28) is available, the Taylor expansion of $g_\theta(\boldsymbol{\theta})$ in Eq. (12) is obtained by evaluating Eq. (5) together with Eq. (11) in the DA framework. Then, the gradient of g_θ is readily obtained by extracting the twelve first order coefficients of its Taylor expansion and is used to compute $\boldsymbol{\alpha}$ using Eq. (13). In addition, similarly to DAMC, each numerical propagation of the standard algorithm is substituted by a polynomial evaluation of Eq. (28). Moreover, accurate methods for the computation of the parameters c_1^i and c_2^i

can be developed since the evaluation of the polynomial approximation of g_θ is fast. More in detail, to further speed up the execution of a DA-based LS simulation, the algorithm described in Section 2.2 is slightly modified. The step **LS** 2(c) is divided in two parts. First, the maximum of the PF and the associated value c^i_{max} for the i-th sample are identified. The two values c^i_1 and c^i_2 are then computed only if the maximum of the PF is positive. This is done by applying a secant method using two initial guesses that are close to c^i_{max}, one slightly larger and the other slightly smaller. If the maximum of the PF is negative there is no need to compute intersections and $P^{1D,i} = 0$. Although this approach increases the complexity of the method, it avoids unnecessary polynomials evaluations increasing the robustness and efficiency of the DA-based LS. The acronym DALS-k is used in the following to indicate the DA-based Line Sampling with expansion order k.

DASS. Similarly to DAMC, in the DA-based Subset Simulation, the numerical propagations at step **SS** 2 are replaced by the fast evaluation of the polynomial map in Eq. (28). All other steps involve sorting and generation of samples through MCMC and do not require any modification. Hereafter this algorithm will be labelled as DASS-k, where k is again the order of the expansion.

4. Numerical Examples

In this section, the performances of the proposed approaches are assessed on the computation of collision probabilities for close encounters in LEO and GEO. All computations are performed on a Intel Core i5 2500 @3.30GHz, 8Gb RAM processor running Sabayon Linux 10 (kernel version 3.5.0).

At first, the DA algorithms are compared against analytical methods and standard Monte Carlo simulations, using test cases in which the relative motion can be considered linear, non-linear and almost-linear, respectively. The test cases are taken from Alfano (2009), where simple Keplerian dynamics is used to compare a set of different methods for collision probability computation. The aim of this analysis is to validate the proposed methods and assess their performances in terms of accuracy and efficiency.

Then, the methods are tested using the high fidelity numerical propagator AIDA. The covariance matrices for each object are obtained after a pseudo orbit determination process, where observed states are obtained with TLEs propagation through SGP4/SDP4. The orbit determination is performed as

a batch least-square optimization, yielding a full 6×6 covariance matrix. The goal of this analysis is to test the methods in real scenarios and assess their reliability.

Since the considered methods employs different number of samples, two figures of merit are used for comparisons (Zio and Pedroni, 2009b). The first figure of merit is the unitary coefficient of variation, Δ, and is defined as

$$\Delta = \frac{\sigma}{\hat{P}_c}\sqrt{N_T}, \tag{29}$$

where σ is the standard deviation of the collision probability from its estimated value $\hat{P}_c$, and N_T is the total number of samples used. The unitary c.o.v. does not depend on the number of samples, since for Monte Carlo methods $\sigma \propto 1/\sqrt{N_T}$. It is designed to enable the comparison of the different methods in terms of accuracy and number of samples required to reach that accuracy level. The lower is the value of Δ, the lower is the variability of the corresponding failure probability and, as a consequence, the higher is the efficiency of the method.

The second figure of merit, Ω, involves both variance and computational time t_c and does not depend on the number of samples N_T as well. It is defined as

$$\Omega = \frac{1}{\sigma^2\ t_c}. \tag{30}$$

It is a measure of the computational efficiency and failure probability variability. The higher the value the higher is the efficiency of the method.

4.1. Validation of DA-based methods

In this section the methods for collision probability computation DAMC, DALS, and DASS are validated against an analytical method and standard Monte Carlo simulation. Three test cases are considered, one with linear relative motion between the two objects, one at the boundary of linear relative motion, and one with nonlinear relative motion. These are respectively test case 5, 6, and 7 of Alfano (2009). The same labelling is used in this paper to ease comparison. The orbital state and covariance matrix are propagated using Keplerian dynamics (Shepperd, 1985) for the standard Monte Carlo method and Alfano's method. In this example, the Taylor expansion of the DCA given in Eq. (28) is then based on the propagation of Kepler's dynamics in DA environment.

The three conjunctions analyzed are detailed in Table 1, where the time, distance, and relative velocity at the closest approach, Δv_{TCA}, are listed. The reference value for collision probability, computed using a standard Monte Carlo simulation is given. For each trial, two sets of initial conditions are sampled from each initial covariance matrix and the associated DCA is searched in the proximity of the nominal TCA. The number of samples N_T for the computation of the reference P_c is selected to achieve a confidence level of 95% and 1% relative error according to Eq. (10). The collision probability obtained using Alfano's formula and its associated percentage relative error with respect to the reference collision probability are given in the last two columns.

Table 1: Time, distance, and velocity of closest approach for the Keplerian test cases and reference value for collision probability, computed with a standard MC method using N_T samples. The collision probability computed with Alfano's approach is given together with the relative error with respect to the reference P_c.

Test case	TCA [days]	DCA [m]	Δv_{TCA} [m/s]	D [m]	P_c (Monte Carlo)	N_T (Dagum)	P_c (Alfano)	% err [-]
5	2.0	2.449	0.520	10	4.454E-02	2.30E+06	4.440E-02	-0.32%
6	2.0	2.449	0.173	10	4.340E-03	2.50E+07	4.324E-03	-0.36%
7	2.0	3.183	0.196	10	1.614E-04	6.71E+08	1.580E-04	-2.13%

The collision probability is then computed using the three DA-based methods and a standard Monte Carlo method. For DAMC and the standard Monte Carlo computations the number of samples is now selected to guarantee a relative error of 5% and a confidence level of 95%. The results are listed in Table 2, where for DAMC, DALS, and DASS the expansion order was set to $k = 3$. The relative error with respect to the reference P_c, number of samples used, computational time, coefficient of variation δ, and figures of merit Δ and Ω are also given.

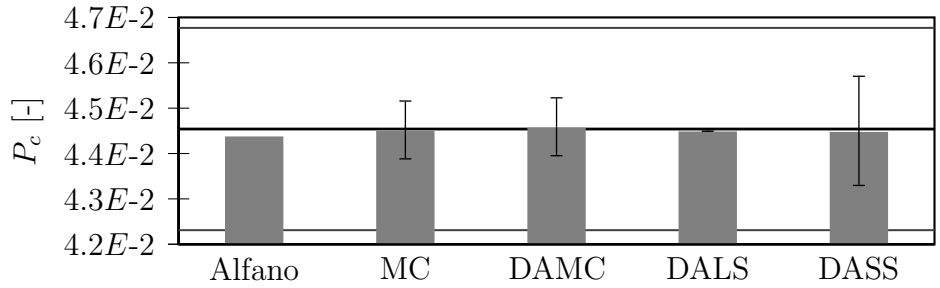
The collision probability values are always in good accordance with the reference value. In particular, the relative error for test case 7 is lower than the one obtained with Alfano's method, since nonlinear effects are captured by using a third-order polynomial approximation. The computational time of the DA-based methods is always lower than standard Monte Carlo. Among all methods DALS is the one that has the lowest Δ and the highest Ω, thus resulting to be the most efficient method. A comparison of the different methods is given in Figure 6. The collision probabilities are plotted as bars

Table 2: Computed collision probability for the Keplerian test cases. For each simulation the relative error with respect to the reference P_c, the number of samples used, the computational time t_c, coefficient of variation δ, and figures of merit Δ and Ω are listed.

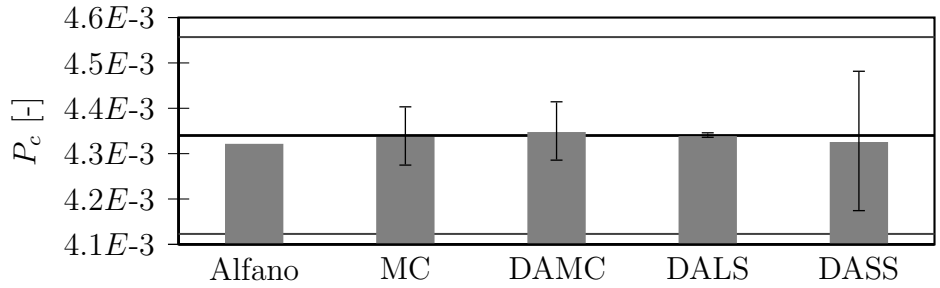
Test case	Method	P_c [-]	% err [-]	N_T	t_c [s]	δ [-]	Δ [-]	Ω [-]
5	MC	4.452E-2	-0.05%	1.0E+5	4.75	1.465E-2	4.63	4.949E+05
	DAMC-3	4.459E-2	+0.11%	1.0E+5	0.67	1.464E-2	4.63	3.503E+06
	DALS-3	4.451E-2	-0.07%	5.0E+3	2.53	7.662E-4	0.05	3.399E+08
	DASS-3	4.450E-2	-0.09%	2.0E+4	0.13	2.738E-2	3.87	5.183E+06
6	MC	4.339E-3	-0.01%	1.0E+6	43.21	1.515E-2	15.14	5.357E+06
	DAMC-3	4.350E-3	+0.24%	1.0E+6	6.67	1.513E-2	15.13	3.462E+07
	DALS-3	4.341E-3	+0.03%	5.0E+3	2.58	1.484E-3	0.11	9.340E+09
	DASS-3	4.328E-3	-0.27%	4.0E+4	0.27	3.586E-2	7.17	1.539E+08
7	MC	1.615E-4	+0.04%	2.7E+7	1155.36	1.514E-2	78.68	1.447E+08
	DAMC-3	1.612E-4	-0.15%	2.7E+7	179.34	1.516E-2	78.76	9.341E+08
	DALS-3	1.621E-4	+0.41%	5.0E+3	1.43	1.936E-2	1.37	7.103E+10
	DASS-3	1.626E-4	+0.72%	6.0E+4	0.43	4.580E-2	11.22	4.193E+10

together with their 1-σ error. The reference probability value is represented by a solid black line and the 5% relative error lines are reported as two solid grey lines. For cases 5 and 6, DALS and DASS show the lowest and highest standard deviations of the collision probability, respectively. Standard Monte Carlo and DAMC provides similar results in terms of P_c and variance for all three test cases. In Figure 6(c) it can be observed that Alfano's method underestimates the collision probability for test case 7, when nonlinearities are relevant. The other methods instead are much more closer to the reference probability.

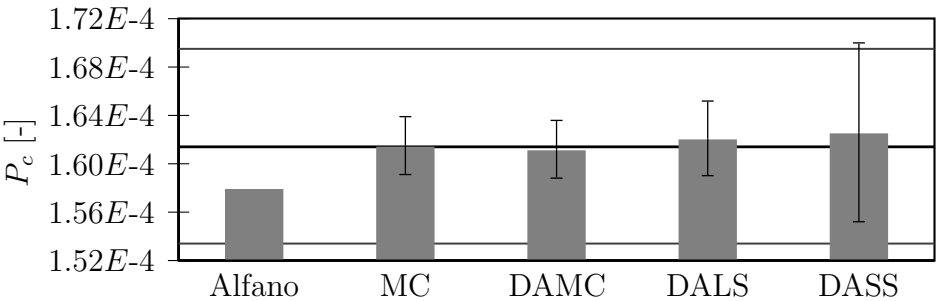
The computational time for the three DA-based methods is plotted in Figure 7, normalized by the t_c obtained with the standard Monte Carlo method to highlight the computational gain. For each of the three DA-based methods a different marker is used: squares for DAMC, circles for DALS, and triangles for DASS. The computational time is plotted for expansion orders ranging from $k = 1$ to $k = 4$ and the markers are coloured accordingly using a gray scale, where black is used for $k = 1$ and light gray for $k = 4$. The computational time is usually lower than the one of the standard Monte Carlo method and increases with the expansion order for all methods. Using DAMC, the computational time can be reduced by a factor of 10 with an expansion order up to $k = 3$. Note that the computational gain is



(a) Case 5



(b) Case 6



(c) Case 7

Figure 6: Comparison of the collision probability obtained with the tested methods for the Keplerian test cases. The DA expansion order is $k = 3$ for DAMC, DALS, and DASS. The solid black line is the reference value for collision probability and the gray lines are the 5% relative error bounds.

limited in this case as a simple dynamical model is used, thus there is only a little advantage when pointwise propagations are substituted by polynomial evaluations of Eq. (28).

For the higher probability value (test case 5) the computational time of the three DA-based methods are comparable and the one of DALS is even a bit higher than the one of DAMC and DASS, due to its higher complexity.

In particular, it is worth noting that for order $k = 4$ it also exceeds the computational time of pointwise MC, that for test case 5 requires a lower number of samples. Nevertheless, the computational effort of DALS and DASS decreases for lower collision probability, becoming nearly 10^3 times lower than the one of a standard Monte Carlo method for test case 7.

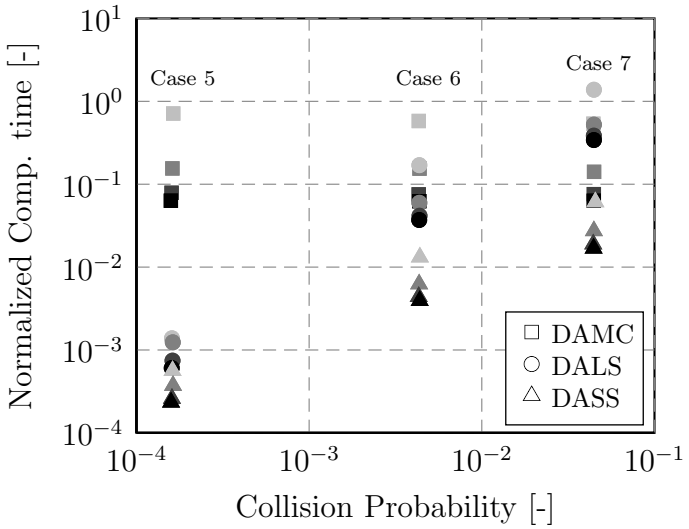
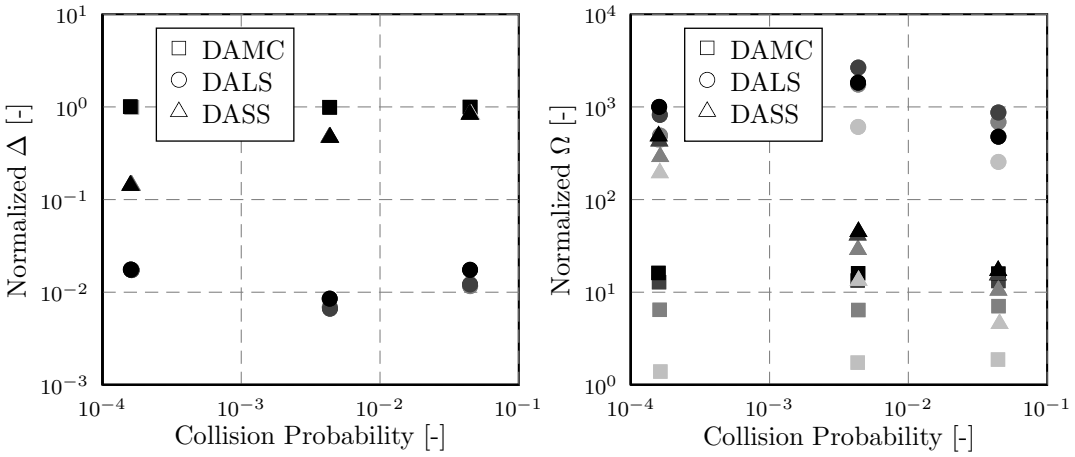


Figure 7: Normalized computational time of DAMC, DALS, and DASS for the Keplerian test cases vs. collision probability for different expansion orders. Markers are coloured using a grayscale, where black is used for the expansion order $k = 1$ and lighter gray for $k = 4$.

The figure of merit Δ, normalized for each test case with the value of the standard Monte Carlo method, is plotted against the collision probability P_c in Figure 8(a). The same criteria used in Figure 7 for markers colouring and shape is used, i.e. different markers are used for each method and they are coloured according to the expansion order using a gray scale. The normalized unitary c.o.v. is equal to 1 for the DAMC since the same number of samples of the standard Monte Carlo method is used. The lower value is achieved with DALS, which is two order of magnitude lower than DAMC. The efficiency of DASS increases for lower probabilities. The use of different expansion orders does not affect the final value of the normalized Δ, since points are overlapping and indistinguishable. The only exception is found for DALS in cases 5 and 6, where the normalized Δ is slightly higher when $k = 1$. This is probably due to a slightly lower accuracy of the first-order DCA expansion

in those cases.



(a) Normalized figure of merit Δ

(b) Normalized figure of merit Ω

Figure 8: Normalized figures of merit Δ and Ω of DAMC, DALS, and DASS for the Keplerian test cases vs. collision probability for different expansion orders. Markers are colored using a gray scale, where black is used for the expansion order $k = 1$ and lighter gray for $k = 4$.

The figure of merit Ω is plotted versus the collision probability in Figure 8(b), again normalized with respect to the value obtained with the standard Monte Carlo method. Since the computational time increases with the expansion order as shown in Figure 7, the value of Ω decreases for higher order. The best performing among the three methods for the considered test cases is DALS, since the normalized figure of merit is at least 10 times larger than the one of DASS and 10^2 times larger than DAMC. For test case 7, where collision probability is lower, the efficiency of DASS in terms of Ω is higher than the other cases. Note that the value of Ω for an expansion order $k = 1$ is lower than the one obtained with $k = 2$ for DALS for test case 5 and 6. As stated before, the reason is the slightly lower accuracy of the map in this case, that is not mitigated by the lower computational time.

To conclude this analysis, the collision probability computed with Alfano's formula and the DA-based methods with an expansion order $k = 1$ are compared in Table 3. It can be observed that using a first-order DA expansion the percentage relative error is similar to that obtained using Alfano's method for test case 7, where the relative motion is no more linear. The DA-methods at first order are therefore equivalent to Alfano's analytical approximation.

Table 3: Comparison of the collision probability computed with Alfano's method and the DA-based methods for the Keplerian test cases with a DCA expansion of order $k = 1$

Test case	P_c (Alfano)	% err [-]	P_c (DAMC)	% err [-]	P_c (DALS)	% err [-]	P_c (DASS)	% err [-]
5	4.440E-02	-0.32%	4.444E-02	-0.23%	4.448E-02	-0.14%	4.432E-02	-0.50%
6	4.324E-03	-0.36%	4.337E-03	-0.06%	4.332E-03	-0.18%	4.361E-03	+0.49%
7	1.580E-04	-2.13%	1.580E-04	-2.13%	1.568E-04	-2.87%	1.584E-04	-1.88%

4.2. Comparison of the methods on real conjunctions

In this section four test cases are considered to test the algorithms for collision probability computation. The selected test cases include LEO and GEO close encounters with different relative velocity at TCA. The selected test cases are listed in Table 4: the satellites involved in each conjunction case and the associated orbital regimes are listed in the second and third column; the other columns report the TCA, DCA, the relative velocity at TCA, and the collision threshold D used for the computation of P_c. On the last column the collision probability computed using Alfano's formula is listed. The initial orbital states used for orbit propagation with AIDA are listed in Appendix A.

Table 4: TCA and DCA computed for the real conjunctions used as test cases

Test case	Sat. No.	Orbit	TCA [days]	TCA	DCA [m]	Δv_{TCA} [km/s]	D [m]	P_c (Alfano)
A	39152	LEO	2.831	2014 Feb 13 15:08:42	51.0	12.757	10	2.850E-3
	27580	LEO	2.950					
B	27453	LEO	1.820	2013 Nov 22 09:07:47	136.6	11.103	16	3.664E-5
	33692	LEO	1.837					
C	37838	LEO	2.707	2013 Nov 24 06:02:00	75.7	0.327	12	5.218E-3
	37840	LEO	3.909					
D	16199	GEO	1.535	2013 Nov 21 13:55:18	937.1	0.784	15	5.804E-4
	29648	GEO	2.007					

The collision probability is computed with DAMC, DALS, and DASS for each test case, and the results are listed in Table 5. Uncertainties on both position and velocity are considered in these simulations. The variance of the initial positions and velocities are estimated from pseudo observations generated using TLE and SGP4/SDP4 and are given in Appendix A. In Morselli et al. (2014) it was shown that, for similar range of uncertainties,

the error of a third order Taylor expansion of the DCA is lower than the collision threshold for a maximum propagation time of one week.

The number of samples of the DAMC are estimated using Eq. (10) considering a relative error $\varepsilon = 5\%$, whereas the number of samples for DALS is fixed to 5×10^3 and the samples for each conditional level of DASS is 10^4. Although the number of samples is much lower than in DAMC, both DALS and DASS can provide good estimates of the collision probability. Taking the value obtained with DAMC as reference P_c, it is possible to compare the three methods in terms of percentage relative difference. In all cases, the computed collision probabilities differ at most 3% from the DAMC value. The largest difference is obtained for test case B, where the collision probability is lower.

Table 5: Computed collision probability P_c, computational time t_c, and figures of merit Δ and Ω. Percentage relative error is obtained taking DAMC-3 collision probability as reference.

Test case	Method	P_c [-]	% err [-]	N_T	t_c [s]	δ [-]	Δ [-]	Ω [-]
A	DAMC-3	2.869E-3	0.0%	1.5E+6	11.19	1.522E-2	18.64	4.686E+07
	DALS-3	2.891E-3	0.8%	5.0E+3	0.71	2.260E-2	1.60	3.300E+08
	DASS-3	2.875E-3	0.1%	4.0E+4	0.28	3.712E-2	7.42	3.141E+08
B	DAMC-3	3.597E-5	0.0%	1.2E+8	875.88	1.548E-2	166.73	3.682E+09
	DALS-3	3.511E-5	-2.3%	5.0E+3	0.42	3.858E-1	27.28	1.298E+10
	DASS-3	3.674E-5	2.1%	7.0E+4	0.54	4.975E-2	13.16	5.543E+11
C	DAMC-3	5.214E-3	0.0%	8.1E+5	6.06	1.535E-2	13.81	2.577E+07
	DALS-3	5.205E-3	-0.2%	5.0E+3	1.06	1.499E-2	1.06	1.549E+08
	DASS-3	5.190E-3	-0.5%	4.0E+4	0.28	3.541E-2	7.08	1.057E+08
D	DAMC-3	5.751E-4	0.0%	7.3E+6	55.16	1.543E-2	41.67	2.303E+08
	DALS-3	5.783E-4	0.6%	5.0E+3	0.63	3.452E-2	2.44	3.982E+09
	DASS-3	5.799E-4	0.8%	5.0E+4	0.36	4.215E-2	9.43	4.650E+09

The computational time t_c required by DALS and DASS is lower than the one of DAMC. The latter is in turn significantly lower than the one of a standard Monte Carlo method, in which the trajectory of each virtual object is numerically propagated up to the close encounter.

Let us consider the test case A for illustrative purposes. With our implementation of the dynamics, a standard MC simulation with 1.5×10^6 samples would require 1.05×10^8 seconds, as approximately 35 seconds are required to propagate each of the two objects to the TCA. The computational time

of the DA methods is given by the time required to 1) perform the DA integrations 2) compute map (28) and 3) run the algorithms based on polynomial evaluations (labelled as t_c in Table 5). Each of the two third order propagations requires approximately 10 times a pointwise integration, and the DA manipulations to compute the DCA expansion requires only fraction of second (Morselli et al., 2014). Thus, the additional cost associated to DA computations is equivalent to only 20 pointwise numerical propagations and this is irrelevant with respect to the gain obtained by substituting numerical integrations with polynomial evaluations. For test case A, the total computational cost (i.e. including DA orbit propagation, map inversion, and probability computation) of a DAMC-3 run is 492.38 s, which is 5 orders of magnitude less than the time that would be required by a standard MC (note that this value can be further reduced by a more efficient implementation of AIDA propagator). This gain in computational time can be higher for cases with lower computational probability (e.g., test case B) and when DALS and DASS algorithms are used.

Figure 9 summarizes the t_c of the three DA methods. Keeping in mind that the computational time of all the DA methods is orders of magnitude lower than that of standard MC, it can be noted that the computational effort of DAMC increases exponentially for decreasing collision probability, whereas the other two methods have drastically lower variations. The t_c of DASS slightly decreases for increasing collision probability, due to the lower number of conditional levels required to converge. On the contrary, the computational time of DALS increases with collision probability. This is a consequence of the control on the maximum value of the PF on the important direction: for test case B, that has the lowest probability, the relative distance between the two objects is higher and the intersection of the two ellipsoids is very small. As a result, most samples produce a one-dimensional collision probability that is zero since the maximum of the PF is negative and no computation of the intersections c_j^i is required, with a reduction of the computational time.

To conclude, it is worth observing that the computational time of the methods can be further reduced as all methods can be classified as "embarrassingly parallel".

The methods are compared in terms of accuracy and efficiency in Figure 10, where the figures of merit Δ and Ω are plotted against collision probability. For each test case the values of Δ and Ω listed in Table 5 are normalized with respect to the value obtained for the DAMC simulation.

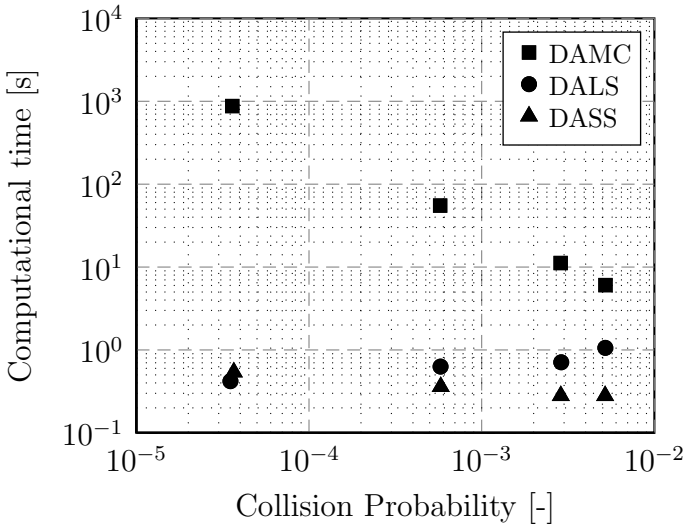


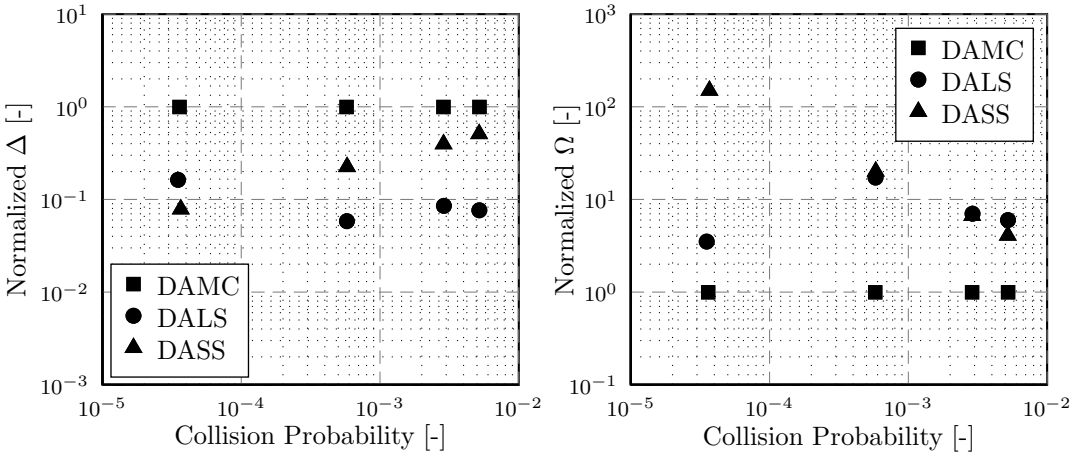
Figure 9: Computational time of DAMC, DALS, and DASS vs. collision probability.

According to Figure 10(a), both advanced methods lead to significant improvements in terms of unitary c.o.v. with respect to DAMC. DALS outperforms DASS for higher probability and its Δ is one order of magnitude lower than DAMC. The performance of DASS increases for lower probability, where it performs better than DALS. The variance of DASS is indeed lower than that of DALS. Nevertheless, DALS shows the lowest computational time.

The figure of merit Ω is plotted against the collision probability in Figure 10(b). For probability higher than 10^{-3}, DALS performs better than DASS and DAMC. The value of Ω exponentially increases for decreasing probability and DASS reaches the same performances of DALS for test case D and outperforms DALS for test case B. The reason is mainly related to the lack of accuracy of DALS for case B, due to the low number of samples for which the one-dimensional probability is non-zero. An higher number of samples should be used with DALS to achieve a more reliable estimate of the collision probability in this case.

4.3. Covariance scaling analysis

The collision probability depends on the shape, size, and orientation of the covariance matrix of the position and velocity at TCA. In this subsection a validation of DALS and DASS for varying initial covariance size is performed,



(a) Normalized unitary c.o.v. Δ

(b) Normalized figure of merit Ω

Figure 10: Performance comparison with normalized figures of merit

using as a reference the values obtained with DAMC. This approach requires that the accuracy of the Taylor expansion of the DCA and TCA is high, with error below 1 meter for all initial covariance sizes. For each test case and covariance size, the accuracy of the Taylor expansion was verified by selecting, from 10^9 samples generated from the full covariance matrix, the 100 with largest displacement from the reference initial state. As the error on the DCA was below the selected threshold in all cases, the value obtained from the DAMC computation corresponds to the one of a pointwise Monte Carlo (for single DCA and TCA conjunctions). Thus, DAMC can be used to validate DASS and DALS.

The principal components of the initial covariance matrices of the two objects, accounting for both position and velocity, are scaled by a factor l and the collision probability is computed. Note that by using principal components it is guaranteed that all components are scaled by the same factor and the correlations coming from the orbit determination process are not altered. The collision probability computation is repeated for different values of the scaling factor l and the results are plotted in Figure 11. A good accordance between the values obtained with DALS and DASS (overlapping solid lines) and the reference DAMC (indicated with squares) is found for all test cases and scale factor l.

Note that all curves show the same behaviour: for smaller initial uncertainties the collision probability is zero or very small, then it increases when

the p.d.f. of the two objects at TCA start to overlap. After reaching its maximum value the collision probability decreases because the volume covered by the p.d.f. continues to grow while the hard-body region (over which the DALS and DASS ideally integrate the combined p.d.f.) remains constant.

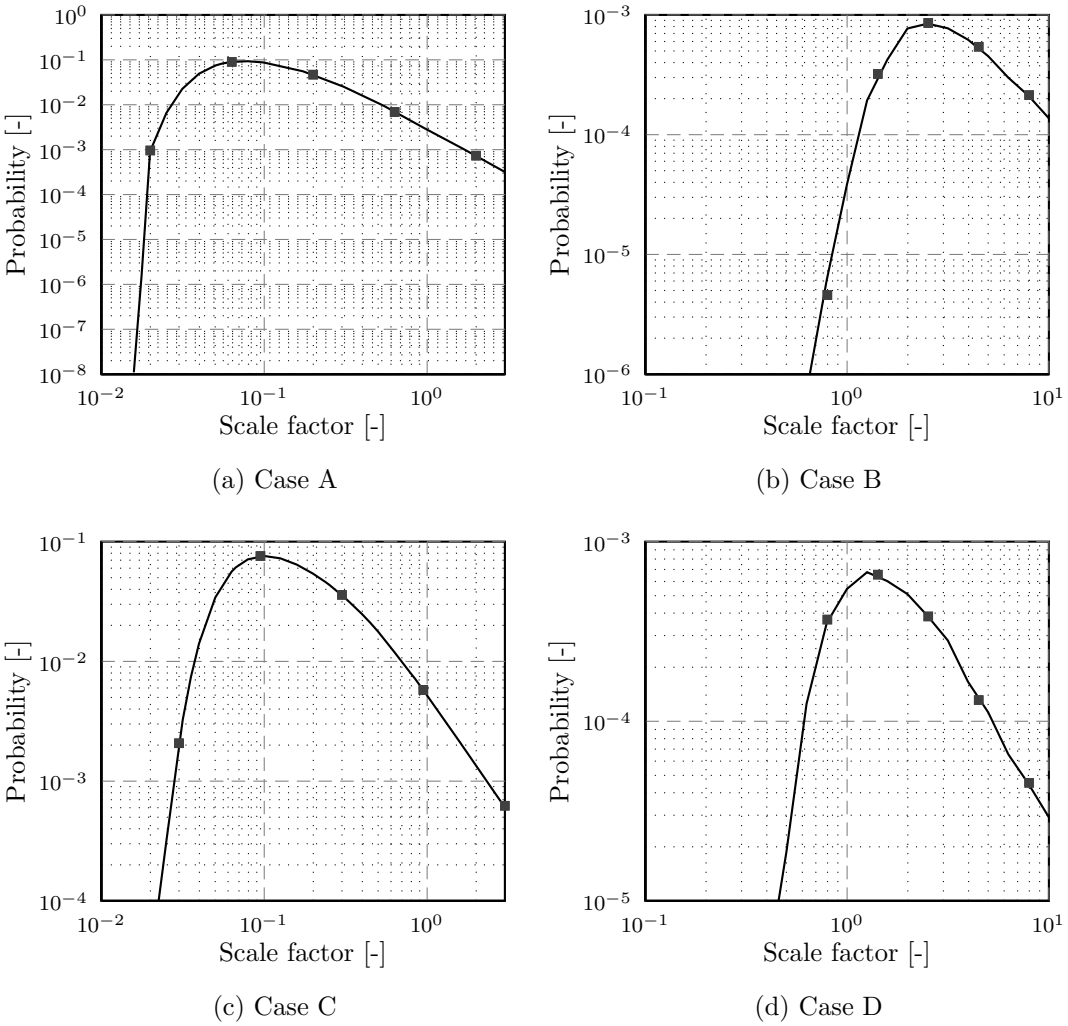


Figure 11: Covariance scaling analysis: collision probability versus scaling factor l. Solid curves are obtained with DALS-3 and DASS-3, squares represent the reference values computed with DAMC-3.

4.4. Non-Gaussian distribution

Besides reducing computational time and managing nonlinearities by setting the expansion order $k > 1$, the DA-based methods can be modified to deal with non-Gaussian distributions. The DA-based propagation with AIDA does not depend on the initial distribution and so are the Taylor expansions of the DCA and TCA. ***Thus, the only requirement for DA-based methods to work with arbitrary initial distributions is to suitably change the sampling procedure.***

The case of a uniform distribution for the initial states is here analyzed and DAMC is used to compute the collision probability. Sampling is performed in principal components, since they are independent. After assembling the 12×12 global covariance matrix, where no correlations between the two colliding objects are considered, the eigenvalues and eigenvectors are computed. The range of each uniform distribution is selected so that its standard deviation σ_i equals the one of the original Gaussian distribution, i.e.

$$\begin{cases} \dfrac{\sqrt{3}}{6\,\sigma_i} & \text{for } x \in \left[-\sigma_i\sqrt{3}; +\sigma_i\sqrt{3}\right] \\ 0 & \text{elsewhere .} \end{cases} \tag{31}$$

The sampled 12-dimensional vector $\boldsymbol{u}$ is then transformed from principal components to J2000 reference frame by

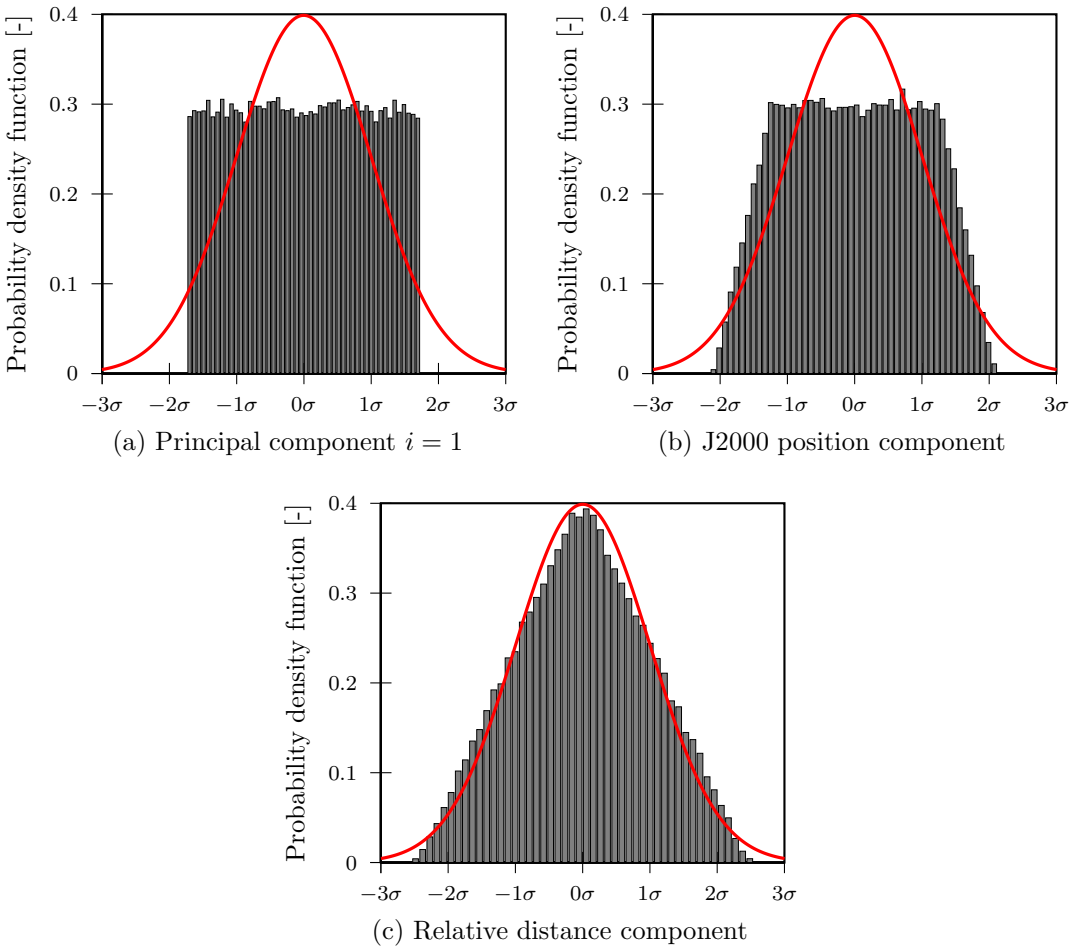
$$\boldsymbol{z} = \boldsymbol{V}\boldsymbol{u} \tag{32}$$

where $\boldsymbol{z} = \{\boldsymbol{x}_0^1; \boldsymbol{x}_0^2\}$ and $\boldsymbol{V}$ is a 12×12 matrix, whose columns are the eigenvectors of the two covariance matrices. The obtained vector $\boldsymbol{z}$ is then used in DAMC algorithm to evaluate the Taylor expansion of the DCA, whose value is then compared with the collision threshold D.

A detail of the resulting p.d.f. throughout the above process is portrayed in Figure 12. It can be observed how the principal component is uniformly distributed and the p.d.f. of the first component of $\boldsymbol{r}_0^1$ in J2000 has a trapezoidal shape. After the propagation, the p.d.f. resembles a triangular distribution.

The computed collision probabilities are listed in Table 6, where the order $k = 3$ is used for the DCA expansion. The collision probability decreases significantly for test case B, whereas it increases by a factor of about 10 for test case C. Smaller variations can be observed for the remaining test cases.

This example shows the importance of considering the proper uncertainty distribution for the initial positions and velocity of the two objects. DAMC,



(a) Principal component $i = 1$

(b) J2000 position component

(c) Relative distance component

Figure 12: Probability distribution function for non-Gaussian initial state uncertainties. The p.d.f. are normalized by the current value of σ and a Gaussian distribution with the same standard deviation is plotted for comparison.

DALS, and DASS have the maximum flexibility in these terms, as they can manage any combination of initial distributions, which can also differ between target and chaser. In addition, the high-order polynomial approximation accounts for nonlinear effects on the p.d.f. resulting from orbital propagation.

5. Conclusion

Three algorithms, based on the high order Taylor expansion of the DCA with respect to initial uncertainties, have been proposed for the computation

Table 6: Collision probability using DAMC-3 and uniform distribution for initial orbital states

Test case	DCA [m]	D [m]	P_c (Gaussian)	P_c (Uniform)	N_T	t_c [s]
A	51.0	10	2.869E-3	2.591E-3	1.7E+6	12.86
B	136.6	16	3.597E-5	4.375E-6	1.2E+8	877.96
C	75.7	12	5.214E-4	4.783E-3	8.9E+5	6.74
D	937.1	15	5.751E-4	7.484E-4	7.5E+6	56.87

of collision probability. This approach enables significant savings in terms of computational effort, since the numerical propagation of the orbital dynamics is replaced by polynomial evaluations. The propagations account for the main sources of perturbation, using up-to-date models for spherical harmonics and air density. The procedure described for the expansion of the TCA and DCA ***, applicable to cases with a single TCA/DCA, *** can be adapted to any set of initial states and using any arbitrary reference frame, which widens the applicability of the method to data coming from any special perturbation catalog. In addition, the probability distribution of the uncertain initial position and velocity is not required to be Gaussian. The algorithm for the identification of the TCA and DCA is insensitive to the initial conditions probability density function. Thus, the only modification to the method described here to work with non-Gaussian distributions is the sampling procedure (as shown by the uniform distribution example).

Besides a DA version of a standard Monte Carlo algorithm, here labelled DAMC, two advanced Monte Carlo methods for the computation of collision probability have been presented and adapted to DA techniques: Line Sampling and Subset Simulation. The resulting methods are named DALS and DASS respectively and both have better performances in terms of computational time and accuracy with respect to DAMC. For collision probability down to 10^{-4}, the DALS outperforms the other two methods, whereas for lower probabilities DASS turns out to be more accurate than the other methods. According to these results DASS is preferable when collision probability is below 10^{-4}, whereas DALS should be used for higher probabilities. The selection of the method to use could be translated in terms of DCA, selecting DALS when the relative distance is below a few hundreds metres and DASS on the other case.

Tests performed on both long-term and short-term encounters have shown that the collision probabilities computed with the three methods are in good accordance. Being based on the Taylor expansion of the TCA and DCA, and since no assumptions are made on the dynamics of the encounter, the presented methods are also suitable for close encounters with low relative velocity, *** provided that a single TCA/DCA occurs in the considered time window.*** It has been shown that, using an expansion order $k > 1$, the methods can capture the effects of nonlinear relative motion on collision probability.

Future studies will focus on further improving the computational performances of the methods. In particular, all three methods can be parallelized with small effort. The codes can be classified as embarrassingly parallel, since the evaluation performed for each sample are independent one from the other. The cases of multiple TCAs and DCAs in long-term encounters will be investigated in future works to allow for the computation of the accumulated nonlinear collision probability. *** In addition, the applicability of the DA-based methods to the case of non-spherical objects will be studied. ***

The applicability of the proposed method to the computation of conflict probability will be also investigated. The method could be directly applied to the case of spherical conflict volumes, whereas the performance function should be properly designed to manage elliptical conflict volumes.

6. Acknowledgment

The authors are grateful to Martin Berz and Kyoko Makino for having introduced us to differential algebra techniques as well as for the implementation of COSY INFINITY, without which this work would not have been possible.

References

Akella, M. R., Alfriend, K. T., Probability of collision between space objects. J. Guid. Control Dyn. 23 (5), 769–772, 2000.

Alfano, S., Addressing Nonlinear Relative Motion for Spacecraft Collision Probability. AAS/AIAA Astrodynamics Specialist Conference, Keystone, CO, 2006a.

Alfano, S., Satellite collision probability enhancements. J. Guid. Control Dyn. 29 (3), 588–592, 2006b.

Alfano, S., Satellite conjunction Monte Carlo analysis. Adv. Astronaut. Sci. 134, 2007–2024, 2009.

Armellin, R., Morselli, A., Di Lizia, P., Lavagna, M., Rigorous computation of orbital conjunctions. Adv. Space Res. 50 (5), 527 – 538, 2012.

Au, S.-K., Beck, J. L., Estimation of small failure probabilities in high dimensions by subset simulation. Probabilist. Eng. Mech. 16 (4), 263–277, 2001.

Bèrend, N., Estimation of the probability of collision between two catalogued orbiting objects. Adv. Space Res. 23 (1), 243–247, 1999.

Berz, M., Modern Map Methods in Particle Beam Physics. Academic Press, New York, 1999.

Berz, M., Makino, K., COSY INFINITY Version 9.1 Programmer's Manual. Michigan State University, East Lansing, MI 48824, MSU Report MSUHEP-101214, 2011. Available on http://www.bt.pa.msu.edu/pub/

Chan, F. K., Spacecraft Collision Probability. Aerospace Press, 2008.

Chan, K., Collision probability analyses for earth orbiting satellites. Adv. Astronaut. Sci. 96, 1033–1048, 1997.

Chan, K., Short-Term vs. Long-Term spacecraft encounters. AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Providence, RI, Paper AIAA-2004-5460, 2004.

Dagum, P., Karp, R., Luby, M., Ross, S., An Optimal Algorithm for Monte Carlo Estimation. SIAM Journal on Computing 29 (5), 1484–1496, March 2000.

de Vries, W. H., Phillion, D. W., Monte Carlo Method for collision probability using 3D satellite models. Advanced Maui Optical and Space Surveillance Technologies Conference 1, 1–11, 2010.

Dolado, J. C., Legendre, P., Garmier, R., Reveling, B., Pena, X., Satellite Collision Probability Computation for Long Term Encounters. Adv. Astronaut. Sci. 142, 2011.

Ghrist, R. W., Plakalovic, D., Impact of Non-Gaussian Error Volumes on Conjunction Assessment Risk Analysis. Proceedings of AIAA/AAS Astrodynamics Specialist Conference, Paper AIAA 2012-4965, 2012.

Jones, B. A., Doostan, A., Satellite collision probability estimation using polynomial chaos expansions. Adv. Space Res. 52 (11), 1860 – 1875, 2013.

Klinkrad, H., Space debris - Models and risk analysis. Springer Berlin Heidelberg, 2006.

Klinkrad, H., Alarcon, J. R., Sanchez, N., Collision avoidance for operational ESA satellites. ESA SP 587, 509–514, 2005.

Koutsourelakis, P. S., Pradlwarter, H. J., Schuëller, G. I., Reliability of structures in high dimensions, part I: algorithms and applications. Probabilist. Eng. Mech. 19 (4), 409–417, 2004.

McKinley, D., Development of a Nonlinear Probability of Collision Tool for the Earth Observing System. AIAA Journal, Paper AIAA-2006-6295, 2006.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E., Equation state calculation by fast computing machines. J. Chem. Phys. 21 (6), 1087–1093, 1953.

Morselli, A., Armellin, R., Di Lizia, P., Bernelli Zazzera, F., A high order method for orbital conjunctions analysis: Sensitivity to initial uncertainties. Adv. Space Res. 53 (3), 490–508, 2014.

Paielli, R. A., Erzberger, H., Conflict probability estimation for free flight. J. Guid. Control Dyn. 20 (3), 588–596, 1997.

Pastel, R., Estimating satellite versus debris collision probability via the adaptive splitting technique. Proceedings of the 3rd International Conference on Computer modeling and simulation, Mumbai, India, 2011.

Patera, R. P., General method for calculating satellite collision probability. J. Guid. Control Dyn. 24 (4), 716–722, 2001.

Patera, R. P., Satellite collision probability for nonlinear relative motion. J. Guid. Control Dyn. 26 (5), 728–733, 2003.

Patera, R. P., Calculating collision probability for arbitrary space-vehicle shapes via numerical quadrature. J. Guid. Control Dyn. 28 (6), 1326–1328, 2005.

Patera, R. P., Collision probability for larger bodies having nonlinear relative motion. J. Guid. Control Dyn. 26 (6), 1468–1471, 2006.

Patera, R. P., Space vehicle conflict-avoidance analysis. J. Guid. Control Dyn. 30 (2), 492–498, 2007a.

Patera, R. P., Space vehicle conflict probability for ellipsoidal conflict volumes. J. Guid. Control Dyn. 30 (6), 1818–1821, 2007b.

Pradlwarter, H. J., Pellissetti, M. F., Schenk, C. A., I., S. G., Kreis, A., Fransen, S., Calvi, A., Realistic and efficient reliability estimation for aerospace structures. Comput. Method Appl. M. 194 (12–16), 1597–1617, 2005.

Righetti, P., Sancho, F., Lazaro, D., Damiano, A., Handling of Conjunction Warnings in EUMETSAT Flight Dynamics. Journal of Aerospace Engineering, Sciences and Applications 3 (2), 39–53, 2011.

Rosenblatt, M., Remarks on a Multivariate Transformation. Ann. Math. Stat. 23 (3), 470–472, 1952.

Sabol, C., Binz, C., Segerman, A., Roe, K., Schumacher Jr, P. W., Probability of Collision with Special Perturbation Dynamics Using the Monte Carlo Method. Adv. Astronaut. Sci. 142, 1081–1094, 2011.

Shepperd, S., Universal keplerian state transition matrix. Celestial Mechanics 35 (2), 129–144, 1985.

Zio, E., Pedroni, N., Estimation of the functional failure probability of a thermal-hydraulic passive system by Subset Simulation. Nucl. Eng. Des. 229 (3), 580–599, 2009a.

Zio, E., Pedroni, N., Functional failure analysis of a thermal-hydraulic passive system by means of Line Sampling. Reliab. Eng. Syst. Safe. 94 (11), 1764–1781, 2009b.

Zuev, K. M., Beck, J. L., Au, S.-K., Katafygiotis, L. S., Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. Comput. Struct. 92, 283 – 296, 2012.

Appendix A. Initial states

Test case A

Satellite ID
39152
Reference UT
10/02/2014 19:12:35.0844445825

# ECI J2000 Position (km)	ECI J2000 Velocity (km/s)
$-3.6439539563E+03$	$+9.1878920823E-01$
$+5.9878758060E+03$	$+5.3316032714E-01$
$+4.9808066441E+00$	$+7.4730343033E+00$

Covariance matrix (km^2, km^2/s, km^2/s^2)

$+9.8237058494E-04$	$+3.8915942674E-04$	$+1.9571722596E-04$	$+1.6609615808E-07$	$-2.2559491098E-07$	$+1.9015667113E-07$
$+3.8915942674E-04$	$+5.7555444198E-04$	$+1.0303424759E-04$	$+4.0954623941E-08$	$-1.5338795992E-07$	$-3.0843253026E-07$
$+1.9571722596E-04$	$+1.0303424759E-04$	$+2.7558675605E-03$	$+1.1746412424E-06$	$-1.9215069558E-06$	$+1.1861317840E-08$
$+1.6609615808E-07$	$+4.0954623941E-08$	$+1.1746412424E-06$	$+1.5490440293E-09$	$-3.1617343607E-10$	$-1.0891636943E-10$
$-2.2559491098E-07$	$-1.5338795992E-07$	$-1.9215069558E-06$	$-3.1617343607E-10$	$+1.8659786917E-09$	$-8.3430324021E-11$
$+1.9015667113E-07$	$-3.0843253026E-07$	$+1.1861317840E-08$	$-1.0891636943E-10$	$-8.3430324021E-11$	$+4.1396922057E-10$

Drag Area to mass ratio (m^2/kg), Cd = 2.2
$+1.2218479923E-02$
SRP Area to mass ratio (m^2/kg), eps = 0.31
$+2.0694123476E-02$
TLE used for nonlinear least square fit
1 39152U 13018C 14041.80040607 .00002412 00000-0 34810-3 0 8423
2 39152 98.0284 121.5048 0018766 38.9612 321.2977 14.77065112 42877

Satellite ID
27580
Reference UT
10/02/2014 16:21:14.1030737758

# ECI J2000 Position (km)	ECI J2000 Velocity (km/s)
$-3.8922004631E+03$	$-6.2512439849E+00$
$-1.0438435353E+03$	$+1.8465268478E-01$
$+5.7962513479E+03$	$-4.1358463997E+00$

Covariance matrix (km^2, km^2/s, km^2/s^2)

$+1.9547374453E-03$	$-9.8049890402E-05$	$+1.0580067225E-03$	$-1.1794923458E-06$	$-2.6119495220E-07$	$+1.3789927232E-06$
$-9.8049890402E-05$	$+1.1188760540E-03$	$+7.8137564847E-05$	$-1.1565590860E-08$	$+1.3391748522E-08$	$-7.1596458005E-08$
$+1.0580067225E-03$	$+7.8137564847E-05$	$+1.0336519655E-03$	$-4.2647489665E-07$	$-1.7889070361E-07$	$+1.1193578612E-06$
$-1.1794923458E-06$	$-1.1565590860E-08$	$-4.2647489665E-07$	$+9.1252134319E-10$	$+8.0369513444E-11$	$-7.6890339138E-10$
$-2.6119495220E-07$	$+1.3391748522E-08$	$-1.7889070361E-07$	$+8.0369513444E-11$	$+1.2941888854E-09$	$-6.9242035260E-11$
$+1.3789927232E-06$	$-7.1596458005E-08$	$+1.1193578612E-06$	$-7.6890339138E-10$	$-6.9242035260E-11$	$+1.4620365188E-09$

Drag Area to mass ratio (m^2/kg), Cd = 2.2
$+1.7201681066E-05$
SRP Area to mass ratio (m^2/kg), eps = 0.31
$+9.8460198665E-02$
TLE used for nonlinear least square fit
1 27580U 01049MY 14041.68141323 .00001359 00000-0 22429-3 0 5945
2 27580 97.8555 3.7377 0047446 330.9850 152.8019 14.71277121622316

Test case B

Satellite ID
27453
Reference UT
20/11/2013 13:26:25.8947440982

# ECI J2000 Position (km)	ECI J2000 Velocity (km/s)
$-2.4979059959E+03$	$-5.9504749387E+00$
$+3.0653803366E+02$	$+3.8129647668E+00$
$+6.7181721934E+03$	$-2.3755680897E+00$

Covariance matrix (km^2, km^2/s, km^2/s^2)

$+1.8615176132E-03$	$-5.9371488217E-04$	$+5.7527411082E-04$	$-6.4855641668E-07$	$+1.4422052340E-07$	$+1.4063005261E-06$
$-5.9371488217E-04$	$+1.3880156120E-03$	$-2.8152360401E-04$	$+3.4638522095E-07$	$-3.9996702959E-08$	$-9.2455468978E-07$
$+5.7527411082E-04$	$-2.8152360401E-04$	$+5.8855927117E-04$	$+7.6250286311E-08$	$-1.6871676008E-07$	$+6.6170167678E-07$
$-6.4855641668E-07$	$+3.4638522095E-07$	$+7.6250286311E-08$	$+7.2731034077E-10$	$+2.4744622174E-10$	$-3.8539559873E-10$
$+1.4422052340E-07$	$-3.9996702959E-08$	$-1.6871676008E-07$	$+2.4744622174E-10$	$+9.1680701357E-10$	$+1.5038111950E-10$
$+1.4063005261E-06$	$-9.2455468978E-07$	$+6.6170167678E-07$	$-3.8539559873E-10$	$+1.5038111950E-10$	$+1.4968859391E-09$

Drag Area to mass ratio (m^2/kg), Cd = 2.2
$+1.0075588399E-03$
SRP Area to mass ratio (m^2/kg), eps = 0.31
$+2.7067261334E-02$
TLE used for nonlinear least square fit
1 27453U 02032A 13324.56002193 .00000354 00000-0 16975-3 0 3501
2 27453 98.3115 330.3370 0012692 70.4075 38.4024 14.24645909592980

Satellite ID
33692
Reference UT
20/11/2013 13:02:7.8791964054

# ECI J2000 Position (km)	ECI J2000 Velocity (km/s)
$+2.7072504230E+03$	$+1.1053867194E+00$
$+6.6185027866E+03$	$-4.5872732420E-01$
$+6.5408784620E+00$	$+7.3812411350E+00$

Covariance matrix (km^2, km^2/s, km^2/s^2)

$+7.4456143946E-04$	$-2.1033418661E-04$	$+1.5630845774E-04$	$-9.5245718864E-08$	$-1.9436808166E-07$	$-9.0622152364E-08$
$-2.1033418661E-04$	$+3.1772267900E-04$	$-6.7843369616E-05$	$+1.9817261436E-09$	$+9.8865423604E-08$	$-2.2049290292E-07$
$+1.5630845774E-04$	$-6.7843369616E-05$	$+1.8524950529E-03$	$-5.4832073914E-07$	$-1.3392393107E-06$	$-2.2891164616E-09$
$-9.5245718864E-08$	$+1.9817261436E-09$	$-5.4832073914E-07$	$+9.1837031296E-10$	$+1.4156627714E-10$	$-9.0630235032E-11$
$-1.9436808166E-07$	$+9.8865423604E-08$	$-1.3392393107E-06$	$+1.4156627714E-10$	$+1.2051410084E-09$	$+3.8737522574E-11$
$-9.0622152364E-08$	$-2.2049290292E-07$	$-2.2891164616E-09$	$-9.0630235032E-11$	$+3.8737522574E-11$	$+2.6817511423E-10$

Drag Area to mass ratio (m^2/kg), Cd = 2.2
$+6.1554384920E-01$
SRP Area to mass ratio (m^2/kg), eps = 0.31
$+2.8921957815E-05$
TLE used for nonlinear least square fit
1 33692U 99025DGD 13324.54314675 .00015012 00000-0 56966-2 0 3993
2 33692 99.1407 67.9439 0028402 336.9020 23.1694 14.30917181292566

Test case C

Satellite ID
37838
Reference UT
21/11/2013 13:04:12.9898434877

# ECI J2000 Position (km)	ECI J2000 Velocity (km/s)
$-2.7064133232E+02$	$-6.9641618091E+00$
$+7.2375327257E+03$	$-2.6533008678E-01$
$+8.8144484510E-01$	$+2.5459684582E+00$

Covariance matrix (km^2, km^2/s, km^2/s^2)

$+1.4620065692E-03$	$+1.5096126850E-04$	$-3.4342152452E-04$	$+3.2500902338E-08$	$+1.1756166169E-06$	$-1.8919723104E-08$
$+1.5096126850E-04$	$+2.1171590357E-04$	$-5.6611199550E-05$	$+1.8808430093E-07$	$+1.5635248049E-07$	$-6.8748866154E-08$
$-3.4342152452E-04$	$-5.6611199550E-05$	$+7.3178022570E-04$	$-1.8310062662E-08$	$-4.5741482493E-07$	$+8.8305361044E-09$
$+3.2500902338E-08$	$+1.8808430093E-07$	$-1.8310062662E-08$	$+2.493095592 8E-10$	$+5.3419271859E-11$	$+1.3816886668E-10$
$+1.1756166169E-06$	$+1.5635248049E-07$	$-4.5741482493E-07$	$+5.3419271859E-11$	$+1.0872884658E-09$	$-1.6378238816E-11$
$-1.8919723104E-08$	$-6.8748866154E-08$	$+8.8305361044E-09$	$+1.3816886668E-10$	$-1.6378238816E-11$	$+5.7539188515E-10$

Drag Area to mass ratio (m^2/kg), Cd = 2.2
$+1.3696048019E-03$
SRP Area to mass ratio (m^2/kg), eps = 0.31
$+1.1868850460E-03$
TLE used for nonlinear least square fit
1 37838U 11058A 13325.54459479 .00000586 00000-0 18131-3 0 5302
2 37838 19.9787 92.3179 0009179 164.4307 195.6381 14.09686725108936

Satellite ID
37840
Reference UT
20/11/2013 08:13:18.4635964036

# ECI J2000 Position (km)	ECI J2000 Velocity (km/s)
$-2.5924255453E+02$	$-6.9640911370E+00$
$+7.2375143678E+03$	$-2.5310140058E-01$
$+8.2501627159E-01$	$+2.5436846038E+00$

Covariance matrix (km^2, km^2/s, km^2/s^2)

$+8.9908619704E-04$	$+3.5482498530E-05$	$-1.5690448148E-04$	$-2.2950305574E-08$	$+6.7406759066E-07$	$-2.1839226852E-09$
$+3.5482498530E-05$	$+1.3622581555E-04$	$-8.5489681255E-06$	$+1.2828235413E-07$	$+3.7833918148E-08$	$-4.7743784534E-08$
$-1.5690448148E-04$	$-8.5489681255E-06$	$+5.2397255801E-04$	$+6.6104495832E-09$	$-2.4509570267E-07$	$+1.1577334597E-09$
$-2.2950305574E-08$	$+1.2828235413E-07$	$+6.6104495832E-09$	$+1.8248705329E-10$	$-8.7155540637E-12$	$+1.1263652901E-10$
$+6.7406759066E-07$	$+3.7833918148E-08$	$-2.4509570267E-07$	$-8.7155540637E-12$	$+6.0575620993E-10$	$-3.1320777230E-13$
$-2.1839226852E-09$	$-4.7743784534E-08$	$+1.1577334597E-09$	$+1.1263652901E-10$	$-3.1320777230E-13$	$+4.4991999269E-10$

Drag Area to mass ratio (m^2/kg), Cd = 2.2
$+1.6733308516E-02$
SRP Area to mass ratio (m^2/kg), eps = 0.31
$+2.6790857074E-02$
TLE used for nonlinear least square fit
1 37840U 11058C 13324.34257481 .00000265 00000-0 00000+0 0 5857
2 37840 19.9633 92.2212 0012886 173.2946 186.7632 14.10655180108878

Test case D

Satellite ID
16199
Reference UT
20/11/2013 01:04:17.9371109605

# ECI J2000 Position (km)	ECI J2000 Velocity (km/s)
$-4.0984596292E+04$	$-7.1565317033E-01$
$+9.3894851315E+03$	$-2.8957835671E+00$
$+2.7224571229E+03$	$-7.5321799039E-01$

Covariance matrix (km^2, km^2/s, km^2/s^2)

$+2.0594892337E-03$	$+1.8379005001E-03$	$+4.7338792975E-04$	$-1.1332545815E-07$	$-9.2599726031E-08$	$-2.3507445814E-08$
$+1.8379005001E-03$	$+8.7961151778E-03$	$+8.4366722845E-04$	$-3.5906104057E-07$	$+9.8838124092E-08$	$+2.8114925162E-08$
$+4.7338792975E-04$	$+8.4366722845E-04$	$+5.4860265792E-03$	$-9.5190527137E-08$	$+2.7889672850E-08$	$+8.1851469305E-09$
$-1.1332545815E-07$	$-3.5906104057E-07$	$-9.5190527137E-08$	$+2.2266514440E-11$	$-3.2210326166E-12$	$-8.0528400269E-13$
$-9.2599726031E-08$	$+9.8838124092E-08$	$+2.7889672850E-08$	$-3.2210326166E-12$	$+1.1764972771E-11$	$-4.2665830474E-12$
$-2.3507445814E-08$	$+2.8114925162E-08$	$+8.1851469305E-09$	$-8.0528400269E-13$	$-4.2665830474E-12$	$+2.7035786499E-11$

SRP Area to mass ratio (m^2/kg), eps = 0.31
$+1.1223807797E-06$
TLE used for nonlinear least square fit
1 16199U 85102A 13324.04465205 -.00000265 00000-0 10000-3 0 2554
2 16199 14.6620 1.3261 0008520 116.5622 48.8663 1.00297344105627

Satellite ID
29648
Reference UT
19/11/2013 13:45:33.4224164486

# ECI J2000 Position (km)	ECI J2000 Velocity (km/s)
$+4.2091373228E+04$	$+1.8931804031E-01$
$-2.5929194691E+03$	$+3.0683059623E+00$
$-9.2251747965E+01$	$+6.6909152476E-04$

Covariance matrix (km^2, km^2/s, km^2/s^2)

$+1.4139689390E-03$	$+5.7834535594E-04$	$-5.2183084638E-05$	$-2.1498800482E-08$	$-1.0394009353E-07$	$+1.3075451281E-11$
$+5.7834535594E-04$	$+7.9198371095E-03$	$-1.5459818308E-04$	$-3.3496996136E-07$	$+1.2443268919E-08$	$+7.3094127670E-10$
$-5.2183084638E-05$	$-1.5459818308E-04$	$+4.5115226302E-03$	$-4.8657365415E-09$	$+4.6785900696E-09$	$+4.9130205025E-11$
$-2.1498800482E-08$	$-3.3496996136E-07$	$-4.8657365415E-09$	$+1.9350129603E-11$	$-1.1375786363E-12$	$+9.1002280124E-15$
$-1.0394009353E-07$	$+1.2443268919E-08$	$+4.6785900696E-09$	$-1.1375786363E-12$	$+8.6020208781E-12$	$-3.4871343335E-15$
$+1.3075451281E-11$	$+7.3094127670E-10$	$+4.9130205025E-11$	$+9.1002280124E-15$	$-3.4871343335E-15$	$+2.3739652425E-11$

SRP Area to mass ratio (m^2/kg), eps = 0.31
$+4.0010579635E-06$
TLE used for nonlinear least square fit
1 29648U 06056A 13323.57330350 -.00000244 00000-0 10000-3 0 1373
2 29648 0.0702 90.3404 0001774 125.3902 140.9346 1.00270688 25513