

A Subset Simulation based technique for calculating the probability of collision

Utkarsh Ranjan Mishra

Texas AM University

Weston Faber

L3Harris Technologies

Suman Chakravorty

Texas AM University

Islam Hussein

Thornton Tomasetti

ABSTRACT

Analytically computed value of Probability of Collision for long term engagements between two space objects using traditional schemes, which only consider some time span around the time of closet approach, can sometimes be incorrect by orders of magnitude. Sampling based methods are presented as a robust alternative to analytical schemes. To decrease the computational burden of simulating a large number of particles, a novel subset simulation based MCMC scheme is introduced to compute in-orbit space-object collision probability.

The collision probability is expressed as a product of larger conditional failure probabilities by introducing intermediate failure events. Well-chosen large (relative to collision probability) values of nested conditional failure probabilities can be estimated by means of simulating only a limited number of samples. The resulting efficiency and accuracy of the suggested scheme are demonstrated against independent benchmarks that use other techniques for calculating the probability of collision.

1. INTRODUCTION

The increase in anthropogenic space objects has lead to the appearance of involuntary space object collisions. The growth of orbital debris from such collisions and debris generated by rocket launches, and the increase in the number of operational satellites particularly because of the miniaturisation of spacecraft systems and deployment of mega constellations has led to more frequent and dangerous space object conjunctions. Estimating and limiting the probability of on-orbit collisions is critical for safe space operations. Multiple collision risk parameters like the distance of closest approach and probability of collision estimates are often employed as the criteria when making collision avoidance manoeuvre decisions. We present a novel formulation for doing conjunction assessment and calculating the probability of collision using Subset Simulation.

Numerous methods exist that compute probability of collision (P_c) when relative motion is assumed to be rectilinear [2, 1]. Essentially the problem boils down to the P_c being equal to the multi-variate integral of Gaussian relative position density over the volume of the collision tube that is swept out by the combined hard-body of the two space objects over a specified time interval [2]. Cases with nonlinear relative motion, typically associated with long-term encounters have been usually effectively tackled using Monte Carlo (MC) methodology [2, 6, 8].

Subset Simulation(SS) is a stochastic simulation method from reliability engineering literature for risk assessment [7, 6, 3]. It uses sampling from conditional distributions using Markov Chain Monte Carlo algorithms for estimation of small probabilities of rare failure events. The core idea of SS is to express a small failure probability as the product of larger conditional probabilities of intermediate failure events, thus converting a rare event simulation problem into simulating a sequence of more frequent nested events. This makes the sampling-based calculation of successive conditional probabilities using MCMC feasible.

The on-orbit collision risk estimation problem is adapted for SS by defining nested intermediate events i.e., $F_1 \supset F_2 \supset \dots \supset F_M$ where $F_M = F$ is the failure event. The intermediate failure events are chosen such that the conditional probabilities $P(F_j|F_{j-1})$ are large. This converts the original problem of evaluating a small failure probability to a sequence of intermediate problems that correspond to the estimation of larger conditional probabilities each one of which can be calculated using MCMC methods. In brief, SS uses MCMC sampling to evaluate the probability that the objects are within d_2 distance (say 5 km) given that the objects are already within d_3 distance (say 10 km). Next, it tries to evaluate the probability that the objects are within d_1 distance (say 2 km) given they are within d_2 distance (5 km) and so on. In this way, it successively telescopes into the region of interest with the effective use of MCMC sampling.

Failure defined as two objects coming closer than some specified cutoff distance (like the combined radius of the two objects) is usually a rare event whose probability can be effectively calculated using Subset Simulation(SS). Analytical schemes that look at failure in only a time-span around time of closest approach account for trajectories that failed in some particular window. Failures at different time steps are not independent events and some probability of collision estimation methods make the mistake of multiple counting i.e., accounting for a failed sample that had already failed in the past. The suggested method samples trajectories of the two objects, at risk of collision, and is free of double-counting thereby giving more meaningful estimates of the probability of collision, closest approach distance and 'engagement window'.

This paper uses the MCMC based Subset Simulation method to evaluate the probability of collision while introducing and addressing following important ideas:

- Considering multiple close encounters vs time-span around the time of closest approach
- Engagement period vs Engagement event
- Double counting of collision probability mass
- Validity of Gaussian-ity assumption when time of closest approach is far into the future and effects of using unscented vs linear propagation schemes.

The following sections also contain details on the MCMC based Subset Simulation (SS) algorithm for calculating the probability of collision. Also included is a simulation against benchmark test cases [2, 6] comparing vanilla MC and Subset Simulation for calculating P_c .

2. PROBABILITY OF COLLISION COMPUTATION USING STOCHASTIC SIMULATIONS

The uncertainties in the orbital elements of space objects can be mapped to a probability of collision via a multi-variate integral over the volume swept out by the combined hard-body area of the two objects normal to the velocity vector. [2, 6, 1].

Many of the methods used for computing this multi-dimensional integral [8, 1] involve restrictive assumptions like neglecting the uncertainties in the velocities and constant uncertainty in position, using Gaussian distributions to represent the uncertainties in states, treating position uncertainties as uncorrelated, and making objects move only along straight lines at constant velocity during conjunction window[6]. These assumptions remain valid when the pdf of the relative position of the two objects remains Gaussian and the relative motion is rectilinear. Improvements have been made to include uncertainty in velocities and long term encounters [4], and mitigate non-Gaussian behavior using local Gaussian and Gaussian Mixture Model representations [10, 5]. The cumulative probability of collision P_c is found by integrating the three-dimensional, Gaussian, relative position density over the collision tube swept out by the combined hard body of the two space objects over a specified time interval (t_i, t_f) .

$$P_c = \frac{1}{\sqrt{8\pi^3}\sigma_x\sigma_y\sigma_z} \iiint_{V(t_i, t_f)} \exp\left[-\frac{x^2}{2\sigma_x^2} + \frac{-y^2}{2\sigma_y^2} + \frac{-z^2}{2\sigma_z^2}\right] dx dy dz \quad (1)$$

The probability density has been written in diagonal frame of the relative position-error covariance matrix for convenience.

There are multiple complications like defining an appropriate integration volume $V(t_i, t_f)$ that need careful consideration in evaluating Equation 1. A major issue with this approach is the assumption about linear relative motion which may not hold in long-term encounters where the collision tube will not be straight invalidating the dimensional reduction used for linear motion. The variances σ_x^2 , σ_y^2 , and σ_z^2 that are used in the P_c calculation can also be different when computed by analytical propagation of state error co-variance while preserving Gaussian-ity versus when computed using particle propagation or even unscented transform. This differences in the actual variances versus the analytically propagated variances becomes more pronounced as the starting time of the time span around the time of closest approach increases. Another possible issue is the possibility of double counting a particle that has failed before t_i and fails again in the time span (t_i, t_f) . Consider Figure 1 that shows a hypothetical scenario with the trajectories of four samples being forward propagated in time. Each sample 1-through-4 comprises of one sample of object-1 and one sample of object 2 respectively. The red circles denote the instances when the objects come within the collision threshold i.e., radius of collision of each other. The green region shows some time-span (t_i, t_f) .

Consider computing the probability of collision in time-span (t_i, t_f) denoted by green region with just these four trajectories generated by forward propagating samples that were generated using Monte Carlo sampling on some distributions of object-1 and object-2. If it is not realised that sample-1 has already failed a long time before t_i and it is meaningless to double count it again in (t_i, t_f) , then the incorrect value of the probability of collision in the time-span (t_i, t_f) would be equal to $\frac{1}{2} = 0.5$. It is clear that the answer should be equal to $\frac{1}{4} = 0.25$ i.e., out of the four samples only one fails so the probability of failure is equal to 0.25. It is borderline impossible to avoid such double counting in analytical schemes but this can be handled easily with particle sampling based methods.

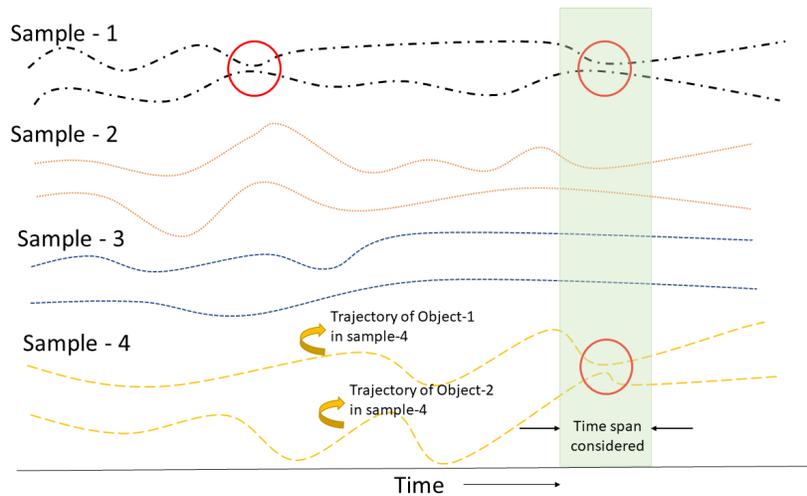


Fig. 1: Counting failed samples : double counting

2.1 Monte Carlo process

Given the probability distribution function of the object at any initial time, Monte Carlo analysis randomly produces samples from the distribution. For P_c calculations, n -points are sampled for both the objects and n -pairs with one sample each from both the objects are created. Both the samples in each of the n -pairs are appropriately propagated for m -time steps. Next, the euclidean distance between the corresponding (in time) propagated states of the two objects is computed to get m dimensional distance vectors corresponding to each pair. If the distance at any time step is lesser than the combined object radius of the two objects a collision is counted. This process is repeated for all the n -samples.

2.2 Subset Simulation method

Subset simulation is a stochastic simulation method to compute small failure probabilities efficiently. The basic idea is to write the probability of failure as the product of larger conditional probabilities. The failure probability is written as:

$$P(F) = P(F_1) \prod_2^n P(F_j | F_{j-1}) \quad (2)$$

where $F_j \subset F_{j-1} \dots \subset F_1$ are nested intermediate failure events, $P(F)$ is the probability of failure event while $P(F_j|F_{j-1})$ is the conditional probability of F_j given F_{j-1} failure region. F_1 i.e., the initial region is initialized using a standard Monte Carlo simulation and defining some cut-off on the performance function of each particle. The samples conditional to the failure region F_1 are computed using a Markov Chain Monte Carlo (MCMC) algorithm. Defining a new cut-off gives a smaller failure region F_2 . This process is repeated till the failure region is identified. Setting cutoffs for the intermediate failure regions can be bypassed by choosing $P(F_j|F_{j-1}) = p_0$. This implicitly sets the $[100 \times (1 - p_0)]^{th}$ percentile value of performance function as the cutoff (C_j).

The cutoff C_j and performance function $g_x^l(\cdot)$ can be used to find if some particle \mathbf{x} lies in conditional level $j + 1$ by checking the following conditions:

$$g_x^l(\mathbf{x}) = \begin{cases} \leq C_j, & \text{iff } \mathbf{x} \text{ lies in } l\text{-th conditional level} \\ > C_j, & \text{iff } \mathbf{x} \text{ does not lie in } l\text{-th conditional level} \end{cases} \quad (3)$$

Since the conditional probability is set to be equal to p_0 at each iteration, the collision probability becomes:

$$P(F) = P(F_n|F_{n-1})p_0^{n-1} \quad (4)$$

which becomes,

$$P(F) = p_0^{n-1} \frac{N_F}{N} \quad (5)$$

where N_f are the number of samples out of N total number of samples at conditional level n that failed i.e had a relative distance less than the collision threshold.

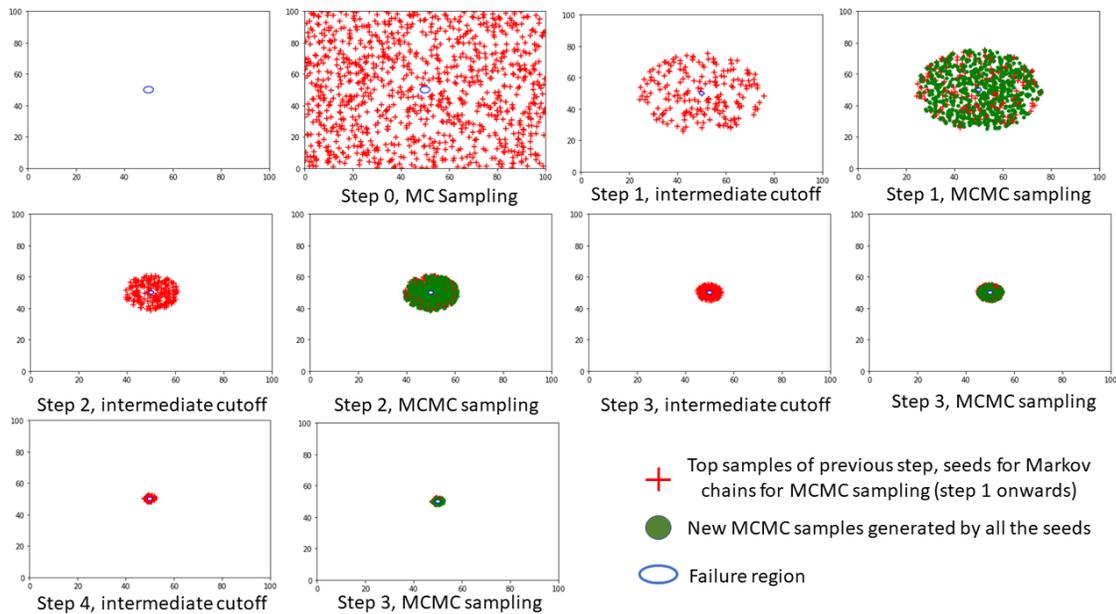


Fig. 2: Toy Example: Subset Simulation Algorithm telescoping into the regions of interest

Figure 2 illustrates the subset simulation in action. Let the state space be two dimensional and both the states be uniformly distributed between zero and one hundred. Let a unit circle centered at (50, 50) be the failure region whose probability mass is to be quantified via sampling. A brute force Monte Carlo method would involve drawing a large number of samples from the 2D distribution and counting the number of samples that land inside the failure region. To get good results a very large number of MC samples would be required. Subset Simulation can outperform MC by selecting samples in appropriate regions of state space by using the sample's value function. Algorithm 1 defines this process in detail. In each step a tighter cutoff is set to get intermediate failure regions. The cutoffs are set on the

output of the value function for each sample. In this case the value function is set to be the distance of the samples from (50,50). Then the samples are drawn from strictly inside the new intermediate failure region. This is done via MCMC sampling with the conditional probability set as the target distribution. This process of setting cutoff and then sampling over conditional distribution is repeated till the stopping criteria is met. In practice this leads to telescoping into the regions of interest and consequently results in getting better estimated value of high dimensional integrals using far less number of samples compared to MC.

Algorithm 1 Subset Simulation Algorithm

Input:

- ▷ p_0 , conditional failure probability;
- ▷ N , number of samples per conditional level;

procedure SUBSET SIMULATION

Set $j = 0$, number of conditional level

Set $N_F(j) = 0$, number of failure samples at level j

Sample $\theta_0^{(1)}, \dots, \theta_0^{(N)} \overset{i.i.d.}{\sim} \pi(\cdot)$

for $i = 1, \dots, N$ **do**

if $g^{(i)} = g(\theta_0^{(i)}) > b$ **then**

$N_f(j) \leftarrow N_f(j) + 1$

end if

end for

while $N_f(j)/N < p_0$ **do**

$j \leftarrow j + 1$

 Sort $\{g^{(i)}\} : g^{(i_1)} \leq g^{(i_2)} \leq g^{(i_3)} \leq \dots \leq g^{(i_N)}$

 Define $b_j = \frac{g^{(i_{N-Np_0})} + g^{(i_{N-Np_0+1})}}{2}$

for $k = 1, \dots, N p_0$ **do**

 Starting from $\theta_j^{(1),k} = \theta_{j-1}^{i_{N-Np_0+k}} \sim \pi(\cdot|F_j)$, generate $1/p_0$ states of a Markov chain $\theta_j^{(1),k}, \dots, \theta_j^{(1/p_0),k} \sim \pi(\cdot|F_j)$ using MMA.

end for

 Renumber: $\{\theta_j^{(i),k}\}_{k=1, i=1}^{Np_0, 1/p_0} \mapsto \theta_j^{(1)}, \dots, \theta_j^{(N)} \sim \pi(\cdot|F_j)$

for $i = 1, \dots, N$ **do**

if $g^{(i)} = g(\theta_0^{(i)}) > b$ **then**

$N_f(j) \leftarrow N_f(j) + 1$

end if

end for

end while

2.3 Markov Chain Monte Carlo

The Markov Chain Monte Carlo(MCMC) method are a class of algorithms that sample from a desired probability distribution by generating a Markov chain that has the desired distribution as its equilibrium distribution and recording the states of this chain. If high enough number of steps are made in the Markov chain, then the distribution of the samples matches the desired distribution. MCMC methods can be used in finding out numerical approximations of multi-dimensional integrals. Algorithm 2[11] describes the steps involved in implementing the Modified Metropolis Algorithm. For use in Subset Simulation the Modified Metropolis Algorithm attempts to pick samples from the following distribution:

$$\pi(\theta|F_n) = \frac{\pi(\theta)I_{F_n}(\theta)}{P(F_n)} \quad (6)$$

where $I_F(\theta)$ is the identifier function that is equal to zero if θ is not in the set F and $I_F(\theta) = 1$ if θ is in the set F .

Algorithm 2 Markov-Chain Monte Carlo Modified Metropolis Algorithm

Input:

- ▷ $\theta^{(1)} \in F$, initial state of a Markov chain;
- ▷ N , total number of states, i.e., samples;
- ▷ $\pi_1(\cdot), \dots, \pi_d(\cdot)$, marginal PDFs of $\theta_1, \dots, \theta_d$, respectively;
- ▷ $S_1(\cdot|\alpha), \dots, S_d(\cdot|\alpha)$, uni-variate proposal PDFs depending on a parameter $\alpha \in R$ and satisfying the symmetry property $S_k(\beta|\alpha) = S_k(\alpha|\beta)$;

procedure MCMC ALGORITHM**for** $i = 1, \dots, N-1$ **do** Generate a candidate state ζ **for** $k = 1, \dots, d$ **do** Sample $\tilde{\zeta}_k \sim S_k(\cdot|\theta_k^{(i)})$ Compute the acceptance ratio, $r = \frac{\pi_k(\tilde{\zeta}_k)}{\pi_k(\theta_k^{(i)})}$ Accept or reject $\tilde{\zeta}_k$ by setting $\tilde{\zeta}_k = \tilde{\zeta}_k$ with probability $\min\{1, r\}$ or $\tilde{\zeta}_k = \theta_k^{(i)}$ with probability $1 - \min\{1, r\}$ **end for** Check whether $\zeta \in F$ by system analysis and accept or reject ζ by setting $\theta^{(i+1)} = \zeta$, if $\zeta \in F$ or $\theta^{(i+1)} = \theta^{(i)}$, if $\zeta \notin F$ **end for****Output:**

- ▷ $\theta^{(1)}, \dots, \theta^{(N)}$, N states of a Markov chain with stationary distribution $\pi(\cdot|F)$
-

2.4 Optimal scaling of proposal distribution

The effectiveness of Subset Simulation is dependent on how quickly and well the Markov chain generated by Markov Chain Monte Carlo (MCMC) algorithm, explores the parameter space and converges to its stationary distribution. The convergence is directly determined by the choice of proposal distribution. The efficiency of MCMC is not sensitive to the type of proposal PDFs but it does depend on the spread. Optimal scaling[9] refers to the parameter tuning required to make the resultant Markov Chain converge to stationary as past as possible. A large proposal distribution would increase the step size but would decrease the acceptance rate because many suggested moves will be rejected due to MCMC stepping criteria while a small proposal distribution would result in higher acceptance rate but poor exploration leading to higher number of steps required for convergence. It has been shown that for optimal sampling the Metropolis algorithm should be tuned accept approximately 23% of the proposed moves.

This paper uses a very simple adaptive optimal scaling for proposal distribution algorithm which sets the proposal distribution to be equal to $k \times P$ where P is the block diagonal matrix constructed from the covariance matrices of two objects. If the acceptance rate is too high or too low then the step is restarted from scratch with a new k . k i.e., the scaling parameter is increased if the acceptance rate is too high and decreased if the acceptance rate is too low.

3. SIMULATION RESULTS

The Subset Simulation method is compared to a standard Monte Carlo simulation using test cases with long engagement period and varying degrees of non-linearity in the relative motion. The test cases are taken from Alfano[2] and use simple Keplerian dynamics for state and state error co-variance propagation to compare the performance of Subset-Simulation method with naive MC methods. A similar exercise to prove the efficacy of Subset Simulations in efficiently recreating results with less number of samples has already been done[6] but restricted itself to a time-span around the time of closest approach and used a differing criteria for optimal scaling of MCMC proposal distribution.

The cases with low probability of collision and long engagement period from Alfano[2] came with not just the states and state error co-variances at t_i (where (t_i, t_f) was the engagement period) but also the initial states and state error covariance at t_0 . The t_i, t_f can be user defined or subject to constraints such as the need to begin and end the time span in a region of very low probability density so that the differences in the shape of collision tubes will be inconsequential in the overall calculation and subsequent comparison. This constraint is enforced by checking if the instantaneous

probability is very near zero at the start and end times. The initial state and state error covariance at t_0 can be used to do a Monte Carlo Simulation.

To investigate the true P_c for a long term engagement for case 7 [2], one hundred thousand samples (each twelve dimensional sample comprises of the sampled state of object one and object two) were generated from the pdfs of the two objects at t_0 . Each sample was forward propagated using simple Keplerian dynamics. A sample fails if the distance of closest approach is less than the radius of collision. Collisions were recorded not just in the arbitrarily defined engagement period (t_i, t_f) but also from t_0 to any t_f . Figure 3 plots the cumulative number of samples for case 7[2] that failed versus time. The red colored strip shows the time span (171300s, 174300s) considered for evaluation of P_c around the time of closest approach 172800s. Looking at this it is clear that a very large majority of the failures happen outside of this time span. The red band in Figure 3 corresponds to the time interval considered by Alfano[2] in evaluating the P_c for case number 7. In this particular case, Alfano reports the P_c calculated using MC with a billion samples to be .000161462, Table 1 reports the P_c for a one million sample MC to be 0.000164. But with merely a 100,000 samples MC, Figure 3 shows that the long term P_c is 1.279e-2 which is two orders of magnitude higher than the reference P_c value obtained when considering only a time span (t_i, t_f) . This shows that such a restrictive time span around time of closest approach may be useful for short term encounters but it completely fails to capture the true P_c for long term encounters.

Note that no double counting was made in creating Figure 3. This means that the failure was recorded only for the first time instance if a particle failed more than once. In the time interval denoted by the red strip in Figure 3, the P_c was equal to 0.00014 i.e. while if double counting is allowed it is equal to 0.0017. Another point to note is that the P_c in some time interval around the time of closest approach might not even be the highest P_c . The taller steps at the start of the plot indicate greater number of particles failed there compared to the time span around the time of closest approach. This raises a question on the utility of using time of closest approach in P_c calculations for long term encounters.

Particle based methods do not suffer from difficulties in removing probability mass of objects that have already failed. But these methods are usually prohibitively computationally expensive. This is where efficient particle sampling based algorithms like Subset Simulations can generate a lot of value.

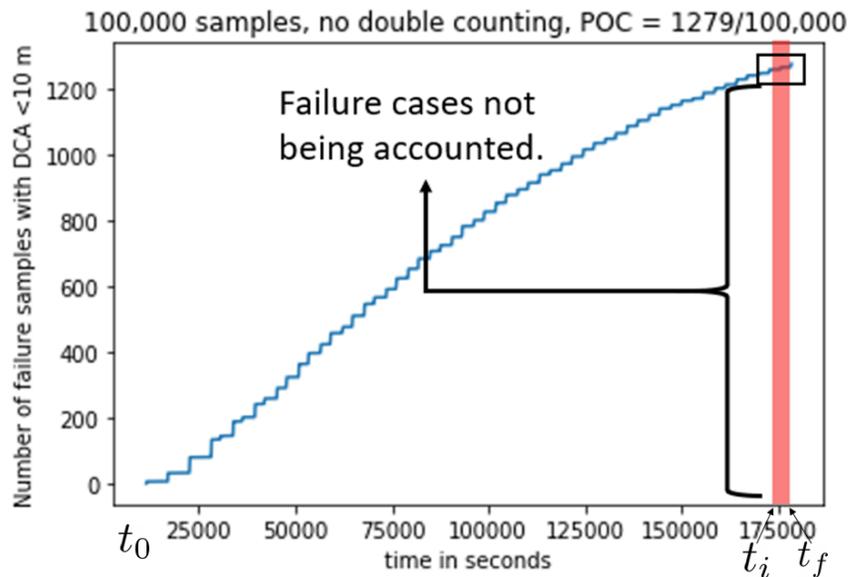


Fig. 3: Case 7[2] : Cumulative number of samples out of 100,000 samples that failed versus time

3.1 Parameters and Setup

Case 5, 6 and 7 were taken from Alfano [2], all three cases involve two space objects in LEO undergoing long term encounter. Case 5 has linear relative motion, case 6 approaches the boundary of linear relative motion, and case 7

involves nonlinear relative motion between the two satellites. Note that although the space objects undergo long term encounter, the P_c calculation shown here for SS and MC is only for a small time interval around the time of closest approach as described by Alfano[2].

To produce results listed in Table 1 and Table 2 the state and state error covariance at time of closest approach given in Alfano[2] were used. Pairs of samples were drawn from the pdfs of the two satellites and forward and backward propagated to cover the time span around the time of closest approach. If at one or many time steps in this time span the relative distance was below the radius of collision then the sample was counted as a failure. The Radius of collision was set at 10m for case 5,6 and 7. For running Subset Simulation $p_0 = 0.25$ and $N = 10,000$ were set fixed at each level. For calculating the reference P_c , 1,000,000,000 samples were used for case 7 and 100,000,000 samples each were used for case 6.

3.2 Results

Table 1 gives the P_c calculated via Subset Simulation using only a fraction ($\sim \frac{1}{15}$) of the number of samples required by MC and was ~ 15 times faster than the MC runs. The SS experiments were repeated for 20 times and mean and standard deviation in the values of P_c thus obtained have also been provided in 2 to give an idea about the spread in the P_c calculated via Subset Simulation.

Case	Reference P_c	MC P_c	MC # samples	SS P_c	SS # samples
5	4.4499e-02	4.4821e-02	1,000,000	4.4398e-02	$3 \times 10,000$
6	4.3005e-03	4.471e-03	1,000,000	4.28019e-03	$6 \times 10,000$
7	1.61462e-04	1.67e-04	1,000,000	1.654785e-4	$7 \times 10,000$

Table 1: P_c calculation using MC sampling and Subset Simulation

Case	Percent error MC	Percent error SS	Mean P_c SS	SD P_c SS	Speed gain
5	0.73	0.227	4.4475e-02	1.8923e-03	32.1X
6	3.965	0.48	4.2525e-03	2.3232e-04	16.2X
7	3.43	2.49	1.60149e-04	1.07e-05	14.78X

Table 2: P_c errors, MC vs SS

4. CONCLUSION

Instead of looking at some time span (t_i, t_f) around time of closest approach, the cumulative number of particles that fail in the time span ($0, t_f$) was investigated. With a simple example, it was shown that the P_c calculations for long term engagements can be incorrect by orders of magnitude if arbitrarily chosen time spans around time of closest approach are used.

The issue of double counting probability mass of already failed samples was discussed and related to how the time span around the time of closest approach might not even have the highest number of sample failures. The need for time-efficient particle sampling based methods was stressed. Finally, a Subset Simulation based P_c calculation method was presented which can reproduce results comparable to MC simulations that use much larger number of samples.

- [1] Maruthi R. Akella and Kyle T. Alfriend. Probability of collision between space objects. *Journal of Guidance, Control, and Dynamics*, 23(5):769–772, 2000.
- [2] S. Alfano. Satellite conjunction monte carlo analysis. *Advances in Astronautical Sciences and Technology*, 134:2007–2024, 2009.
- [3] Siu-Kui Au and Edoardo Patelli. Rare event simulation in finite-infinite dimensional space. *Reliability Engineering System Safety*, 148:67–77, 2016.
- [4] Vincent T. Coppola. Including velocity uncertainty in the probability of collision between space objects. *American Astronautical Society 12-247*.
- [5] Kyle DeMars, Yang Cheng, and Moriba Jah. Collision probability with gaussian mixture orbit uncertainty. *Journal of Guidance, Control, and Dynamics*, 37, 05 2014.

- [6] A. Morselli, R. Armellin, P. D. Lizia, and F. B. Zazzera. A high order method for orbital conjunctions analysis: Monte carlo collision probability computation. *Advances in Space Research*, 1:311–33, 2015.
- [7] I. Papaioannou, W. Betz, K. Zwirgmaier, and D. Straub. Mcmc algorithms for subset simulation. *Probabilistic Engineering Mechanics*, June, 2015.
- [8] R. P. Patera. Satellite collision probability for nonlinear relative motion. *Journal of Guidance, Control, and Dynamics*, 26, 2003.
- [9] Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351 – 367, 2001.
- [10] Christopher Shelton and John Junkins. Probability of collision between space objects including model uncertainty. *Acta Astronautica*, 155, 12 2018.
- [11] Konstantin M. Zuev, James L. Beck, Siu-Kui Au, and Lambros S. Katafygiotis. Bayesian post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions, 2011.