COLLISION RISK ASSESMENT AND AVOIDANCE MANOEUVRES. NEW TOOL CORAM FOR ESA

Ignacio Grande-Olalla[(1)], Noelia Sanchez-Ortiz[(1)], Juan Antonio Pulido[(1)], Klaus Merz[(2)]

[(1)] DEIMOS Space S.L.U., Ronda de Poniente 19, 2°2, Tres Cantos, Madrid, 28760 Spain, email: [ignacio.grande, noelia.sanchez, juan-antonio.pulido]@deimos-space.com
[(2)] ESA/ESOC (Space Debris Office), Robert-Bosch-Str.5,64293 Darmstadt, Germany, email: Klaus.Merz@esa.int

ABSTRACT

This paper presents the CORAM software tool developed by DEIMOS Space for the ESA/ESOC Space Debris Office.

CORAM is designed to help the operator in the task of assessing the collision risk of a conjunction between two objects, and in proposing an optimal avoidance manoeuvre to reduce the collision risk to acceptable levels considering operational constraints.

It is capable of reading different input formats, analysing a time interval and assessing the collision risk of each encounter in that interval. It includes algorithms for conjunctions at low and high relative speed, and the objects can be modelled as simple spheres or as a complex body composed of oriented boxes, following certain attitude laws.

1 INTRODUCTION

CORAM is divided in two different tools.

- CORCOS is the tool responsible for collision risk assessment, input/output of scenario files and propagation.
- CAMOS makes use of CORCOS libraries to compute the optimal avoidance manoeuvre needed by the target satellite to reduce the collision risk (or increase the miss-distance) to a requested level.

In a normal execution, an operator can use CORCOS to evaluate the collision risk of an encounter. If the risk is considered too high, CAMOS can be used to propose avoidance manoeuvres for the operational satellite to reduce the collision risk. The manoeuvres are automatically evaluated with CORCOS so the operator will see the same output as when evaluating the risk.

2 CAPABILITIES

The CORAM SW package is capable of reading the input orbit files in several formats: state vector at an epoch, ephemeris file for an interval, a TLE file or a CSM file.

In addition, it is possible to provide a covariance matrix at an epoch that will be propagated if necessary to use a realistic position uncertainty to calculate the collision risk. Some input formats will provide their own covariance matrix (like the CSM file) or an ESOC in-house look-up table can be used to provide an initial estimation, like in the TLE case.

CORAM is powered by several propagators, which will be selected depending on the input format. A force-model based propagator using a Runge-Kutta 7(8) integrator is used for CSM and state vector, and also for covariance matrix propagation; the well-known SGP4 propagator is used for TLE. Finally a Lagrangian interpolator is used for ephemeris input format.

The force-model propagator can be configured to incorporate several perturbations like Earth geopotencial, third bodies, atmospheric drag, solar radiation pressure, etc.

Both impulsive and low-thrust manoeuvres can be configured by the operator or added by CAMOS during the optimisation process. The force-model propagator can manage these manoeuvres, both for the state vector and for the covariance matrix, with thruster error modelled as an uncertainty in the acceleration and the direction of the manoeuvre, impacting the evolution of the covariance information.

3 ALGORITHMS USED IN THE COLLISION RISK ASSESMENT

Depending on the scenario configured by the user, in particular, the object's geometry (spherical or complex) and the relative speed of the collision, there are several algorithms available to the user.

3.1 Spherical geometry

If both objects are spherical, there are several well-known algorithms that can be used:

- Alfriend & Akella [1], a well-known method to compute collision risk that performs the two-dimensional integration of the hard body projection in the encounter plane.
- Patera's method [3] performs the contour integration of the projection, computing the same

Proc. '6th European Conference on Space Debris'
Darmstadt, Germany, 22–25 April 2013 (ESA SP-723, August 2013)

result in a faster way.

- Maximum Probability, assuming spherical covariance, using the maximum likelihood approach [2]. *Fig. 2* shows the existence of such a maximum for every encounter distance.
- Covariance scaling, where the covariance is scaled for both objects in a given interval and for every scale factor, the covariance is evaluated using the method in [3]. This method preserves the shape and orientation of the covariance matrix of each object and it is useful when the covariance is not well-known.

During CORAM development, these algorithms, and some other finally discarded (Chan, Alfano and Foster), have been tested to check the performance, both in terms of run-time and accuracy (see *Fig. 1*). Additionally, extensive analysis of performance under different conditions of geometry and covariance values was executed (see *Fig. 2*).

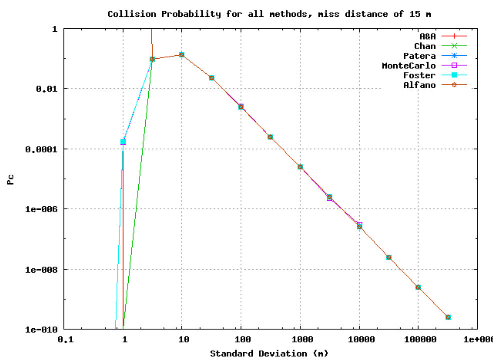

Figure 1: Comparative results of some collision risk algorithms for spherical case.

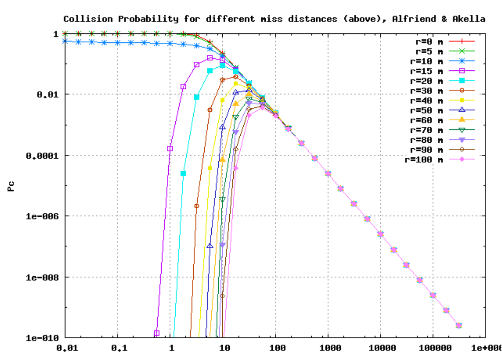

Figure 2: Evaluation of performance for different encounter geometries and orbital accuracy

3.2 Complex geometry

If one of the objects, or both, are complex (composed of oriented boxes), a new method to calculate the collision risk has been devised.

While in the spherical case the hard-body object (collision volume) can be computed as another sphere whose radius is the sum of the radii of the two original spheres, in the complex case this hard body computation is more complicated. It is accomplished by assuming constant attitude and calculating the Minkowski sum [8] of the two objects, and then projecting it onto the encounter plane. Additionally, the collision volume shall be translated to the B-plane, by means of the projection of the vertices of such volume.

The encounter plane is then discretized and sampled. A z-buffer grid [9] is constructed where every cell of the grid is a true/false indicator of the “shadow” of the hard body onto the encounter plane. Every grid contains a small amount of contribution of the collision risk and the last step is to compute the risk associated to every shadowed grid and sum them up.

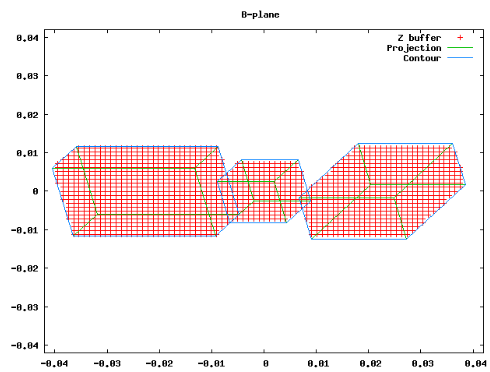

Figure 3: Representation of Encounter plane with the projection of the boxes forming the collision volume (one satellite built by three boxes, and the other based on a unique box), and the Z-buffer evaluation.

The need of considering the actual objects geometry instead of assuming spherical case is very much dependent on the miss-distance and the values of the covariances of the orbital data. Following examples provide some graphical representation of a head-on encounter of two satellites (one satellite built by three boxes, and the other based on a unique box). Geometry of the encounter is shown in *Fig. 4*. Dashed line indicates the equivalent cross-section area assuming spherical geometry whereas the green circle is the nominal encounter point.

In the case of accurate orbital data (small covariance values, about 1 m), the miss-encounter would be perfectly estimated with a high accuracy, only considering the actual geometries of the objects. Otherwise, the integration of the risk along the spherical projection would provide a very low collision risk. This case is represented in *Fig. 5*. The computed collision probability with the complex-geometry algorithm here described is 0.9959, whereas the collision risk computed by algorithms based on spherical assumptions is $1.49 \cdot 10^{-14}$.

In the case of larger uncertainties in the orbital position of the two objects (about 100 m), the probability density function is spread across larger areas of the B-plane, providing very similar results when integrating the risk along the actual object geometries than integrating the risk along the equivalent circle. The computed risk is $2.14 \cdot 10^{-3}$ for the two cases.

3.3 Minkowski sum

To easily compute the Minkowski sum of two complex objects, it is better to divide the objects in convex shapes and compute the sum by pairs, for all combinations and then reconstruct the final object. However, the actual 3D object calculation is not required, only its projection onto the encounter plane. It is possible to skip the 3D reconstruction of the Minkowski sum and calculate the projection directly.

For that, the Minkowski sum is computed for every two boxes (or box-sphere) of the objects but only for the vertex points, without reconstructing any information about the faces. The resulting sum will be also convex.

Those points are then projected onto the encounter plane, and the convex hull that the points form is calculated. This convex hull is the contour of the projected Minkowski sum, represented convex closed irregular polygon.

The entire z-buffer is checked to evaluate what cells of the grid are inside the polygon. Only cells not previously shadowed by other polygon are checked by means of a fast point-in-polygon algorithm.

These steps are repeated for every box-box pair of the complex objects, and the resulting z-buffer grid is evaluated to calculate the collision risk.

In order to do that, it is possible to use Alfriend & Akella or Patera methods on each cell. It can be easily done by replacing every cell by an equivalent circle in the encounter plane and applying a collision risk method to them. The final sum provides the total collision risk.

The z-buffer offers several advantages:

- It is relatively fast.
- It solves the problem of self-shadowing, where different parts of the objects can be accounted several times in the computation of the collision risk. The z-buffer cells have only two states (in shadow / not in shadow) it is not possible to have overlapped sections counting twice.
- It can be easily extended to include other basic shapes, as long as they are convex or could be divided in convex shapes.
- Allows calculating the cross-section of a complex body from a certain point of view, which can be used to estimate the area exposed to atmospheric drag or solar radiation pressure.

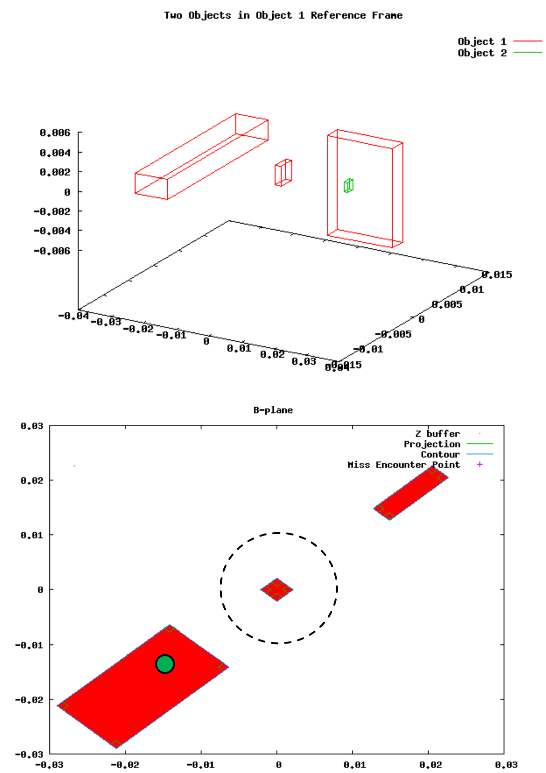

Figure 4: Example Encounter geometry (top figure) and B-plane representation (bottom figure).

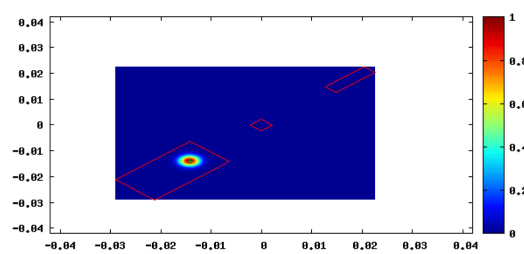

Figure 5: Probability density function for the case of very good orbital data accuracy (~1 m) along the B-plane.

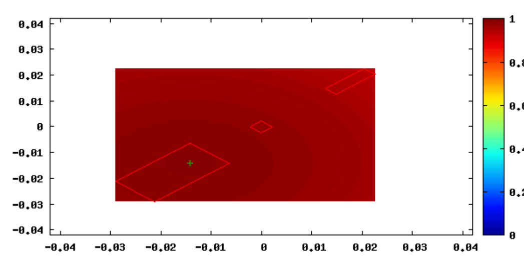

Figure 6: Probability density function for the case of low orbital data accuracy (~100 m) along the B-plane.

3.4 Low-speed encounters

Previously commented methods are in principle applicable only to high-speed encounters, where a linear relative motion and constant orientation, cross section and position uncertainties during the encounter can be assumed.

In a low-speed encounter, however, the conjunction parameters may change in time and it is not possible to evaluate the risk just at the time of closest approach, it is necessary to take into account the whole encounter interval.

An interval-slicing method based on Patera's work [4] has been employed. The method divides the collision interval in slices, and the collision risk is evaluated for every slice. For each slice, the same assumptions as in the high-speed encounter are valid (constant covariance and orientation, linear motion) and the slices can be made as small as necessary for these assumptions to be correct.

To calculate the collision risk of each slice, any other high-speed collision risk algorithm can be used, with a scaling factor to take into account only the contribution of the slice, and not the whole encounter. This means that this method can be used with spherical objects and also for complex geometry objects.

The instantaneous risk (red curve in *Fig. 7*) computed at each slice may be larger than the final computed risk along the interval, since it accounts at every slice as if the out-of B-plane component of the miss-distance is null. Once this fact is properly accounted to evaluate the instantaneous Pc rate (green curve), the cumulated risk can be derived.

3.5 Monte Carlo

In addition to the analytical or semi-analytical methods described previously, CORAM can also simulate the encounter using a Monte Carlo approach, valid for low-speed and high-speed encounters and with any geometry combination. This simulation, however, is much slower than other methods and the main use is to check the results of other methods or to avoid the propagation of the covariance matrix. An example of Monte Carlo use in included in *Fig. 1* with the rest of algorithms.

The collision detection problem involving complex geometries has been solved using the separating axis test [5], a very fast test valid for arbitrarily oriented boxes.

The user may select the number of steps for the Monte Carlo simulation, or alternatively, the user may configure the accuracy and confidence value to estimate the number of runs automatically.

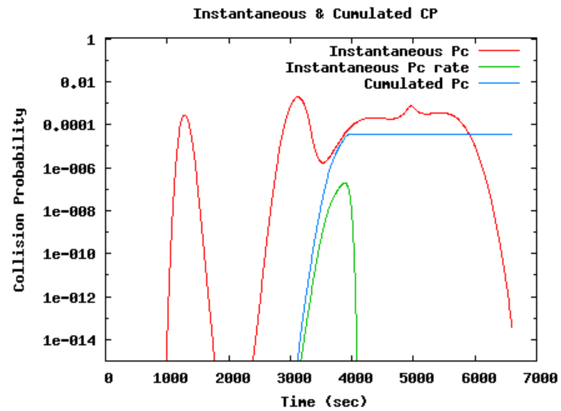

Figure 7: Example of accumulated collision probability along an encounter interval in the case of low-speed.

4 OPTIMAL AVOIDANCE MANOEUVRE COMPUTATION

The computation of the optimal avoidance manoeuvre is performed by CAMOS, the SW utility devoted to manoeuvre optimisation. CAMOS uses most of the functionalities developed for CORCOS:

- Trajectory initialisation (state vector and covariance).
- Orbit acceleration modelling and propagation.
- Object properties initialisation.
- Encounter time search and refinement.
- Collision risk computation, both for low and high speed encounters. Only the analytical methods are used, due to the requirements of the optimisation algorithm described in the following paragraphs. Monte Carlo cannot be used by CAMOS, while complex geometries can be used only if the probability function is evaluated just as output (not as cost function or constraint).

Operationally, CAMOS is usually run once a close encounter between two objects has been analysed by CORCOS, and the obtained collision risk is high enough to deserve the study of an avoidance strategy.

CAMOS can be run in two modes:

- Parametric analysis mode. This mode can assess one or several strategy analyses, where *strategy analysis* should be understood as a one-dimensional or two dimensional parametric execution of a manoeuvre optimisation problem. This mode allows the user to evaluate, e.g., the effect of the manoeuvre execution time on the collision risk, with optimised manoeuvre direction for each selected value of the manoeuvre execution time in the grid. As example, *Fig. 8* shows the effect of a 1-cm/s manoeuvre on the distance of closest approach (DCA) as function of the execution time.

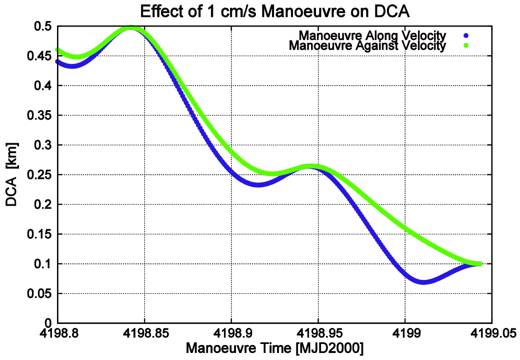

Figure 8: Example of parametric analysis results

- Evaluation mode. It runs just one case within one strategy, and produces specific output files to allow CORCOS to evaluate the selected case (with the newly designed manoeuvres) with risk computation methods not available to CAMOS. The user will usually run CAMOS in evaluation mode for the most interesting case or cases found by running CAMOS previously in parametric mode. Only one case can be evaluated at a time. In addition, this mode can produce optional information on the evolution vs. time of certain trajectory functions, like longitude, latitude, eclipse or location over the South Atlantic region. As example, *Fig. 9* shows the evolution of the longitude of a GEO satellite.

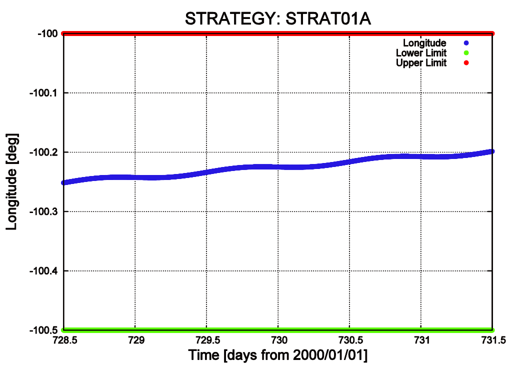

Figure 9: Example of evaluation mode results

In both execution modes CAMOS produces tailored gnuplot scripts for the representation of the obtained data.

The configuration of each strategy analysis is flexible:

- Manoeuvres can be modelled as impulsive or not impulsive
- Manoeuvre directions can be provided in different reference systems:
  - Mean Earth equator of epoch J2000.0
  - True equator and equinox of date
  - Mean equator and equinox of date
  - Local orbital (radial, in-track, cross-track)
  - Local intrinsic (along-velocity, momentum, binormal)
- Each manoeuvre parameter (manoeuvre central time, size, azimuth and elevation) can be defined as fixed, a parameter of the strategy analysis, or an optimisation parameter
- Bounds can be set on manoeuvre parameters, and specific direction constraints can be configured
- Within the optimisation, the cost function can be selected as the collision risk, total delta-V or distance of closest approach (separation vector modulus, or its projection in along-track, cross-track or radial direction)
- Constraints can be set-up in the resulting trajectory: longitude and latitude for GEO satellites, and orbital period and ground track drift for LEO satellites.

CAMOS uses a gradient optimisation package called OPTGRA (see [7]), developed by ESA/ESOC/Flight Dynamics, to find the optimum manoeuvre parameters in each configured problem. The algorithm can deal with equality and inequality constraints. It looks for the optimum solution by moving the initial optimisation parameters tangential to the constraints, and in the direction of steepest descent of the cost function.

Since gradient methods are local optimisation techniques, the solutions found by the algorithm must be understood as local optima and, therefore, must be analysed critically by the analyst in search of the global optimum. For example, manoeuvre execution times have an effect on collision risk that can have a certain sinusoidal component (with its period equal to the orbital period). In that case, the gradient optimisation algorithm would select the local optimum closest to the initial manoeuvre time. In any case, since the tool allows analysing several strategies in one run, each with different selection of strategy or optimisation parameters, the presence of such local optima can be investigated by selecting the manoeuvre time as a strategy parameter instead of an optimisation parameter.

5 OUTPUT FROM CORAM

Output from CORAM includes a set of text files with all the relevant information to aid the operator to evaluate the encounter and plan any necessary action. In particular, an extensive summary file analyses every encounter found with information about the collision risk, covariance matrices, encounter geometry and information about each object.

In addition to the summary file, a set of data files are created, depending on the scenario analysed, that may include gnuplot scripts to plot encounter geometry during a period of time for low-speed encounters, graphical representations of the b-plane, z-buffer and complex geometries in space.

6 TEST CASES

Two cases are provided in this section. The first one for the case of two spherical objects and a high speed encounter is intended to show the general capabilities of CORCOS and CAMOS. The second one is an example of a low-speed and complex geometry scenario.

6.1 Complete Test Case for Spherical Objects and High Speed Encounter

This section shows a test case of CORCOS and CAMOS capabilities. A collision scenario is presented, and CORCOS is used to calculate its properties. Then CAMOS will compute an optimal avoidance manoeuvre, showing the strategy analyses and optimisation settings, and finally CORCOS will re-analyse the new scenario.

A perpendicular, high-speed encounter between two LEO spherical satellites is analysed, where a chaser body on a MEO polar orbit (i=90 deg, R_{per}=8000 km, R_{apo}=10400 km) approaches a target in an equatorial orbit with R_{per}=8000.01 km and same apogee. The trajectory data is defined by state and covariance files at TCA and a very simple propagation model, only central gravity potential, is used for simplicity in the test case.

6.1.1 CORCOS initial assessment

The input files used by CORCOS and CAMOS define the state vectors and covariances of both objects, and configure the scenario. In this case, the orbits are the same but perpendicular, having the same period, and they are defined at TCA. This means that there will be a close approach every half orbit.

The analysis done by CORCOS computes all the encounters in the analysed period of time, shown in *Tab. 1*. Four encounters are found, two per orbit.

Table 1: Initial collision risk assessment of the test case

Time since initial epoch	Miss distance	Collision risk
77 min	52 m	$5.17 \cdot 10^{-5}$
154 min	103 m	$6.74 \cdot 10^{-5}$
231 min	117 m	$5.07 \cdot 10^{-5}$
308 min	206 m	$1.39 \cdot 10^{-6}$

6.1.2 CAMOS optimization

In order to show the capabilities of CAMOS, two different strategy analyses are presented hereafter. Both cases deal with the same encounter event, namely the first one described in *Tab. 1*.

The first strategy is a one-dimensional parametric run with a 10-cm/s manoeuvre, using the manoeuvre execution time as analysis parameter. Two sub-strategies are analysed, with the initial manoeuvre direction along and against the velocity. *Fig. 10* and *Fig. 11* show the results of the analysis on the DCA and probability of collision (PoC) respectively. In both cases the manoeuvre direction is optimised to minimise the PoC.

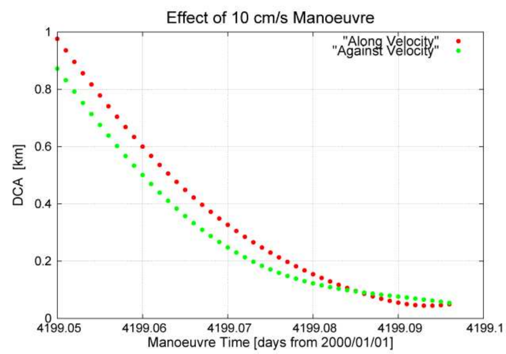

Figure 10: DCA as function of manoeuvre time

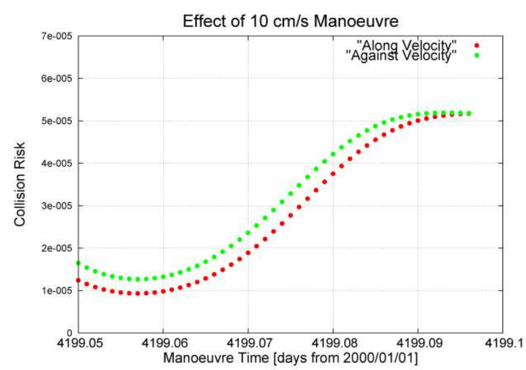

Figure 11: PoC as function of manoeuvre time

The second strategy analyses the effect of a 10 cm/s manoeuvre located in the most favourable point shown in *Fig. 11* (around 4199.055) to investigate the presence of local optima in the azimuth-elevation grid. *Fig. 14* shows the results, confirming that the along and against-velocity directions are the local optima.

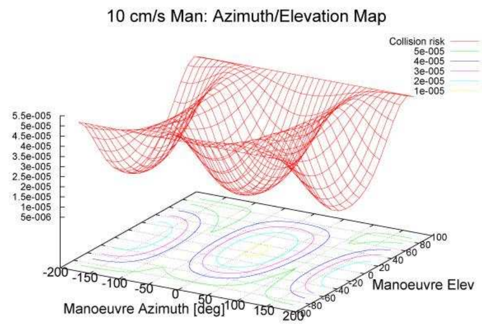

Figure 12: PoC as function of manoeuvre azimuth & elevation

The impact on the risk of the subsequent encounters can also be analysed with CAMOS. *Fig. 13* provides the risk at each encounter as a function of the manoeuvre time. It can be seen how the risk of the two encounters at the opposite point in the orbit (namely 2 and 4) to that which is intended to be reduced (encounter 1) increases for some manoeuvring interval.

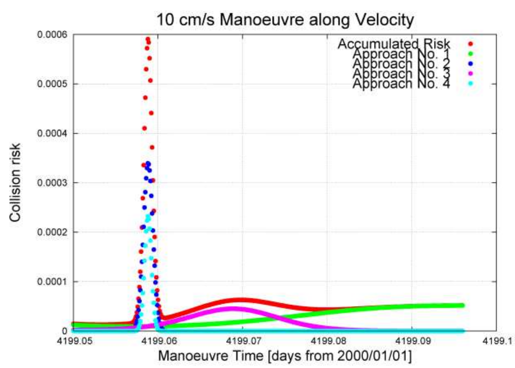

Figure 13: Risk for the different Encounters in the case of an along-track manoeuvre

6.2 Test Case for Complex Geometries and Low Speed Encounters

This test case is limited to CORCOS, as CAMOS capacities remain unchanged no matter the type of object geometry and/or relative velocity of the encounter.

The test case is related to two close-to-GEO orbits with a minor relative inclination (0.025 degrees). One object is composed by three boxes, while the other is made of one unique box. The dynamics makes the two objects to approach twice per orbit, as shown in *Fig. 14*. This figure provides the miss-distance and collision probability as computed by CORCOS for the estimated encounters. Twenty encounters are found in a ten-day time interval.

As already mentioned in section 3.4, the collision probability for low speed encounters is computed by accumulating and scaling the collision probability along each slice of the encounter interval. The resulting probability for all the identified encounters is provided in the bottom plot of *Fig. 14*.

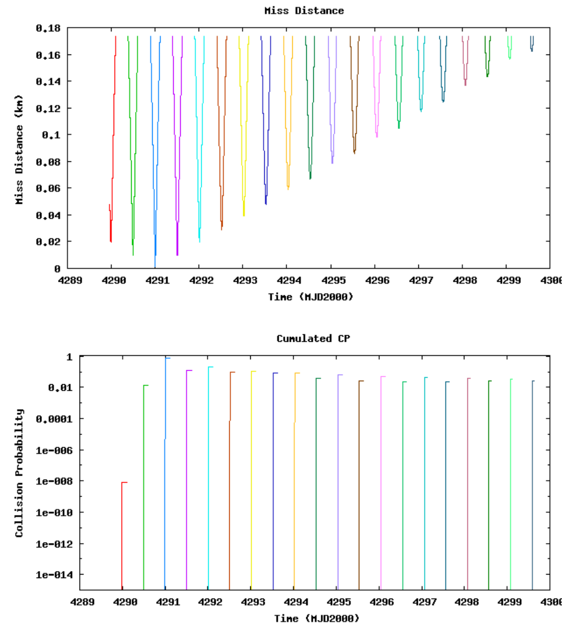

Figure 14: Miss-distance and Collision Probability along the different encounters identified by CORCOS

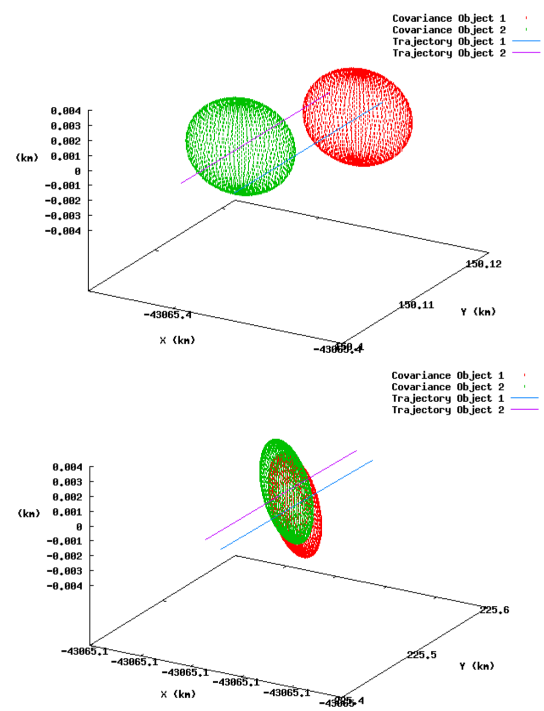

Figure 15: Covariance Ellipsoids at one example slice along the second (top) and fourth (bottom) encounters

It is important to note the differences in encounter geometry and the knowledge of the orbits along the different encounters. The second encounter (green curves) is related to a very low miss-distance (similar to the fourth encounter, pink curve), but the collision probability associated to these two events is almost one order of magnitude different. The geometry of the encounter is different and also the projected density function on the B-plane is very different, mainly due to the dispersion of the uncertainties of the orbits (see *Fig. 16*). In the fourth event, the uncertainty has spread up to a level which makes the integration over the projected area larger than in the second event.

From the top plot in *Fig. 16* one could expect a lower risk, but it has to be considered that the risk at that event is the cumulated risk along the encounter interval. The risk along the two intervals is shown in *Fig. 17*.

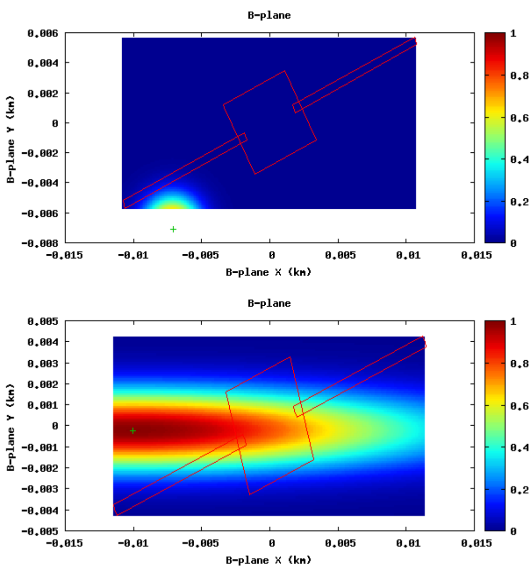

Figure 16: Density Function over the B-plane of an intermediate slice along the second (top) and fourth (bottom) encounters

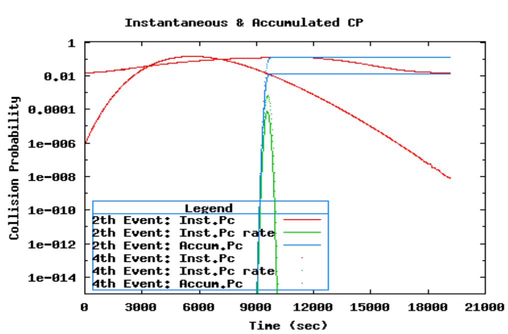

Figure 17: Instantaneous and accumulated collision probability along the second (continuous lines) and fourth (dashed lines) encounter

7 SUMMARY

This paper outlines the capabilities of CORAM as a new tool available to ESA/ESOC Space Debris Office to help the assessment of collision risk and to devise optimal avoidance strategies to reduce the risk to acceptable levels.

The introduction of complex geometries to model the object shape is interesting if the orbit determination processes lead to small covariances and if the satellites have irregular shapes, e.g. large solar panels. In such cases the collision risk can be significantly different when compared to spherical objects.

The ability to calculate the collision risk on low relative speed encounters allows a precise estimation of collision risk in scenarios like nearly co-orbiting conjunction partners, co-located geosynchronous satellites, formation flying or approach manoeuvres.

Finally, the process of risk assessment is complemented with the computation of a set of avoidance manoeuvres. Several optimization strategies with user-configurable constraints allow the operator to choose the best manoeuvre depending on the situation and the satellite's mission requirements.

8 REFERENCES

1. M.R. Akella, K.T. Alfriend. (2000) "*Probability of collision between space objects*", Journal of Guidance,Control and Dynamics, Vol.23, No.5, pp.769-772.

2. Chan, K. (2008), "*Spacecraft Collision Probability*" Aerospace Press.

3. Patera, R. (2001), "*General Method for Calculating Satellite Collision Probability*". Journal of Guidance, Control & Dynamics, vol.24:716-722.

4. Patera, R. (2003). "*Satellite Collision Probability for Nonlinear Relative Motion*". Journal for Guidance, Control & Dynamics, vol. 26, No. 5, 728-733.

5. Ericson, Christer (2006). "*Real-time collision detection*", Elsevier.

6. Sánchez-Ortiz, et al. (2013). "*Computation of Satellite Cross Section in ESA DRAMA tool*", 6A.P-20 presented in this Conference.

7. J.A. Pulido-Cobo, J. Schoenmaekers (2000). "*The Gradient Method Adapted for SMART-1Trajectory Optimisation*", ESOC/Flight Dynamics Technical Note.

8. *http://en.wikipedia.org/wiki/Minkowski_addition*. Checked on 21-04-2013.

9. *http://en.wikipedia.org/wiki/Z-buffering*. Checked on 21-04-2013.