Software Development Kit: Two-Dimension Probability of Collision (Pc) Calculations

CONJUNCTION ASSESSMENT AND RISK ANALYSIS (CARA) PROGRAM

Astrorum, Inc
Waco, Tx

Omitron, Inc
Colorado Springs, CO

National Aeronautics and
Space Administration
Goddard Space Flight Center,
Greenbelt, MD 20771

December 2019

Preface

This document outlines the Two-Dimension Probability of Collision (Pc) and associated software algorithms submitted as part of the Software Development Kit (SDK). The SDK is intended to provide both industry and government customers with a code base with which to perform standard calculations inherent to the Collision Avoidance (CA) problem and as outlined in the CA Standard.

Table of Contents

List of Figures

List of Tables

1.0 Introduction

The CARA Software Development Kit (SDK) contains entries and artifacts for each major algorithm needed to perform the required Collision Avoidance (CA) calculations outlined in the CA Standard. For a typical algorithm, the SDK will include a version of the algorithm, a driver program to take information from a text formatted CDM and execute the algorithm, producing the needed calculation or output, and a series of test cases that exercise the algorithm and produce validated results.

This document describes a series of algorithms inherent to the Pc Omnibus tool, their associated inputs and outputs, the methodology used within each algorithm and examples of usage.

1.1 Required Software

The following list is of software and hardware requirements for use of this SDK:

- Matlab 2016b

2.0 Software Algorithms

2.1 2D Probability of Collision Calculation

One method that may be employed as a method of determining the probability of collision is by transforming a close approach event from a three dimensional problem to a two dimensional problem, which greatly simplifies the calculation of the probability of collision. This calculation is widely used to characterize and analyze close approach events and determine resultant probabilities of collision as a result of mitigation actions.

2.1.1 2D Probability of Collision Foster Function – Mathematical Formulas

The two-dimensional probability of collision is calculated using the method proposed by Foster and Estes[i]. The probability of collision problem is a problem existing in a three dimensional space representing the relative position and velocity vectors of the objects in question, as well as their associated uncertainties. The complexity of this problem, however, may be reduced to a two dimensional problem by assuming rectilinear motion of the two objects during the encounter time. Representing the problem in two dimensions is done by mapping an encounter between two objects onto a two dimensional plane defined by being perpendicular to the relative velocities between the two objects; this is acceptable as the combined uncertainty along the relative velocity vector has no bearing on the calculation of the probability of collision:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \dfrac{\vec{r}_1 - \vec{r}_2}{|\vec{r}_1 - \vec{r}_2|} \\ \dfrac{\vec{v}_1 - \vec{v}_2}{|\vec{v}_1 - \vec{v}_2|} \\ \dfrac{(\vec{r}_1 - \vec{r}_2) \times (\vec{v}_1 - \vec{v}_2)}{|(\vec{r}_1 - \vec{r}_2) \times (\vec{v}_1 - \vec{v}_2)|} \end{bmatrix}$$

The probability of collision can then be represented using only the *x* and *z* components in the new coordinate frame and the geometry and covariance information in the *y* components can be disregarded, greatly simplifying this problem.

Once the problem has been re-characterized in the two dimensional plane, the spatial density of the debris relative to the space vehicle can be characterized as:

$$f(\vec{r}) = \frac{1}{2\pi\sqrt{|\boldsymbol{C}|}} e^{-\frac{1}{2}(\vec{\boldsymbol{r}}-\vec{r}_d)^T \boldsymbol{C}^{-1}(\vec{\boldsymbol{r}}-\vec{r}_d)}$$

Where $\boldsymbol{C}$ is the combined position uncertainties for both objects, $\vec{\boldsymbol{r}}$ is a position on the collision plane, and $\vec{r}_d$ is the debris object's position on the conjunction plane.

$$\vec{\boldsymbol{r}} = \begin{bmatrix} x \\ z \end{bmatrix}$$

$$\vec{r}_d = \begin{bmatrix} x_1 - x_2 \\ z_1 - z_2 \end{bmatrix}$$

From this spatial density, the probability of collision may be calculated by integrating the spatial density over the area of the keep out region as defined by the hard body radius.

$$P_C = \frac{1}{2\pi\sqrt{|\boldsymbol{C}|}} \int_{\oplus} e^{-\frac{1}{2}(\vec{\boldsymbol{r}}-\vec{r}_d)^T \boldsymbol{C}^{-1}(\vec{\boldsymbol{r}}-\vec{r}_d)} \, d\vec{r}$$

This calculation becomes the following for the given methods of representing the cross sectional area using an input hard body radius:

Circular Cross Section

$$P_C = \frac{1}{2\pi\sqrt{|\boldsymbol{C}|}} \int_{-HBR}^{HBR} \int_{-\sqrt{HBR^2-x^2}}^{\sqrt{HBR^2-x^2}} e^{-\frac{1}{2}(\vec{\boldsymbol{r}}-\vec{r}_d)^T \boldsymbol{C}^{-1}(\vec{\boldsymbol{r}}-\vec{r}_d)} \, d\vec{z} \, d\vec{x}$$

Square Cross Section

$$P_C = \frac{1}{2\pi\sqrt{|\boldsymbol{C}|}} \int_{-HBR}^{HBR} \int_{-HBR}^{HBR} e^{-\frac{1}{2}(\vec{\boldsymbol{r}}-\vec{r}_d)^T \boldsymbol{C}^{-1}(\vec{\boldsymbol{r}}-\vec{r}_d)} \, d\vec{z} \, d\vec{x}$$

Square Equivalent Area Cross Section

$$P_C = \frac{1}{2\pi\sqrt{|\boldsymbol{C}|}} \int_{-\frac{HBR\sqrt{\pi}}{2}}^{\frac{HBR\sqrt{\pi}}{2}} \int_{-\frac{HBR\sqrt{\pi}}{2}}^{\frac{HBR\sqrt{\pi}}{2}} e^{-\frac{1}{2}(\vec{\boldsymbol{r}}-\vec{r}_d)^T \boldsymbol{C}^{-1}(\vec{\boldsymbol{r}}-\vec{r}_d)} \, d\vec{z} \, d\vec{x}$$

2.1.2 2D Probability of Collision Foster Function – Source Code Description

The primary function contained within the SDK used for estimating the 2D Probability of Collision of a close approach event using the Foster methodology is the:

Pc2D_Foster.m

routine, which estimates the probability of collision using the formula above.

As inputs, the routine accepts the following:

Table 1: 2D Probability of Collision Foster Routine Input Parameters

Input Variable	Definition
r1	[3X1] ECI Position Vector of the Primary Object (meters)
v1	[3X1] ECI Velocity Vector of the Primary Object (meters/second)
cov1	[6X6] Primary State covariance matrix corresponding to input primary object reference frame
r2	[3X1] ECI Position Vector of the Secondary Object (meters)

v2	[3X1] ECI Velocity Vector of the Secondary Object (meters/second)
cov2	[6X6] Secondary State covariance matrix corresponding to input primary object reference frame
HBR	Combined hard body radius or exclusion zone of the two objects (m)
RelTol	Relative Tolerance used for double integration convergence (1E-08 is recommended)
HBRType	Definition of hard body region, typically “circle”. Allowable inputs: • “circle” – Hard body region defined as a sphere or circle • “square” – Hard body region defined as a cube or square • “squareEquArea” – Hard body region defined as a square with equivalent area to a circle with radius as defined y HBR

The 2D Probability of Collision routine outputs the following:

Table 2: 2D Probability of Collision Foster Routine Output Parameters

Output Variable	Definition
Pc	Probability of Collision calculated using Foster approximation in two dimensional space
Arem	[2X2] Combined covariance projected onto xz-plane in the relative encounter frame.
IsPosDef	Binary flag indicating if the combined, marginalized and remediated covariance has a negative eigenvalue. If the test fails, the Pc is not computed. The function returns NaN for Pc. (Success = 1 & Fail = 0)
IsRemediated	Binary Flag indicating if the combined and marginalized covariance was remediated

Validation cases for this algorithm are contained within the unit test suite for the SDK at:

..\TwoDimensionalPc\UnitTest\ProbabilityOfCollisionCode\Pc2D_Foster_UnitTest.m

These test cases were developed using previously defined stressing cases developed by Alfano 2009[ii], previously existing test cases developed by Omitron to test specific methods of cross-sectional area representation, and a test case designed to test error catching of non-positive definite covariance matrices.

Table 3: 2D Probability of Collision Foster Function Unit Test Cases

Test ID	Description
test01	Alfano test case 1
test02	Alfano test case 2
test03	Alfano test case 3
test04	Alfano test case 4
test05	Alfano test case 5
test06	Alfano test case 6
test07	Alfano test case 7
test08	Alfano test case 8
test09	Alfano test case 9
test10	Alfano test case 10
test11	Alfano test case 11
test12	Omitron test case designed to test circular cross-sectional area calculation of probability of collision
test13	Omitron test case designed to test square cross-sectional area calculation of probability of collision

test14	Omitron test case designed to test square equivalent area cross-sectional area calculation of probability of collision
test15	Omitron test case representing a real event where the covariance matrix was non-positive definite (algorithm remediates covariance and returns a Pc estimate of “0”)

2.1.3 2D Probability of Collision Elrod Function – Mathematical Formulas

The two-dimensional probability of collision can alternatively be calculated using the method proposed by Elrod[iii] which is computationally less intensive but follows the same 2D integration assumptions and formulation as given above. This method differs from the method previously proposed by Alfano[iv] in two distinct ways: this method utilizes the Cholesky decomposition in place of the spectral decomposition to factor the covariance matrix, and this method leverages Chebyshev quadrature in place of the midpoint rule for numerical integration. The formulation of this function begins with the Pc for a circular exclusion zone as given above:

$$P_C = \frac{1}{2\pi\sqrt{|\boldsymbol{C}|}} \int_{-HBR}^{HBR} \int_{-\sqrt{HBR^2-x^2}}^{\sqrt{HBR^2-x^2}} e^{-\frac{1}{2}(\vec{\boldsymbol{r}}-\vec{r}_d)^T \boldsymbol{C}^{-1}(\vec{\boldsymbol{r}}-\vec{r}_d)}\, d\vec{z}\, d\vec{x}$$

This equation is then reduced to a one-dimensional integral by first performing a change in variables of *z, x* to *y, w*. Where *y,w* each have independent standard normal distributions. To accomplish this, the covariance in the conjunction plane, *C,* is converted to a product of upper triangular matrices using a Cholesky decomposition so that:

$$UU^T = C$$

Where:

$$U^{-1}\begin{bmatrix} z \\ x \end{bmatrix} = \begin{bmatrix} y \\ w \end{bmatrix}$$

And:

$$U = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}$$

Applying this transformation of variables to the equation above reduces the portion of the equation in the exponent:

$$(\vec{\boldsymbol{r}} - \vec{r}_d)^T \boldsymbol{C}^{-1}(\vec{\boldsymbol{r}} - \vec{r}_d) \xrightarrow{yields} y^2 + w^2$$

And the function within the integral to:

$$f(y, w) = \frac{1}{2\pi} e^{-\frac{y^2+w^2}{2}}$$

Next the limits of integration must be revised for this transformation of variables, the bounds on *w,* are the more extreme bounds and hence the integration bounds become:

$$w = \pm \frac{HBR}{U_{22}}$$

The bounds on *y* are less straightforward but may be found using the quadratic formula to be:

$$y = \frac{x_0 - U_{12}w \pm \sqrt{HBR^2 - U_{22}^2 w^2}}{U_{11}}$$

The probability of collision calculation then becomes:

$$P_C = \frac{1}{2\pi}\int_{-\frac{HBR}{U_{22}}}^{\frac{HBR}{U_{22}}}\int_{\frac{x_0-U_{12}w-\sqrt{HBR^2-U_{22}^2w^2}}{U_{11}}}^{\frac{x_0-U_{12}w+\sqrt{HBR^2-U_{22}^2w^2}}{U_{11}}} e^{-\frac{y^2+w^2}{2}}\,d\vec{y}\,d\vec{w}$$

Which can be rewritten as:

$$P_C = \frac{1}{\sqrt{2\pi}}\int_{-\frac{HBR}{U_{22}}}^{\frac{HBR}{U_{22}}} e^{-\frac{w^2}{2}}\int_{\frac{x_0-U_{12}w-\sqrt{HBR^2-U_{22}^2w^2}}{U_{11}}}^{\frac{x_0-U_{12}w+\sqrt{HBR^2-U_{22}^2w^2}}{U_{11}}} \frac{1}{\sqrt{2\pi}}e^{-\frac{y^2}{2}}\,d\vec{y}\,d\vec{w}$$

Using the standard normal CDF relations, $\Phi(x) = \frac{1}{2}\left(2 - erfc\left(\frac{x}{\sqrt{2}}\right)\right)$ this can be reduced to a single dimension integration (the complement of the error function is used for better accuracy in estimating small values):

$$P_C = \frac{1}{\sqrt{2\pi}}\int_{-\frac{HBR}{U_{22}}}^{\frac{HBR}{U_{22}}} e^{-\frac{w^2}{2}}\left[erfc\left(\frac{x_0 - U_{12}w + \sqrt{HBR^2 - U_{22}^2w^2}}{U_{11}}\right) - erfc\left(\frac{x_0 - U_{12}w - \sqrt{HBR^2 - U_{22}^2w^2}}{U_{11}}\right)\right]d\vec{w}$$

To facilitate computational speed, this can be approximated as a Chebyshev polynomial with higher degrees resulting in more accurate results:

$$P_C = \sum_{1+N/2}^{N} \omega_i e^{\left(-\frac{\left(\frac{HBRn_i}{U_{22}}\right)^2}{2}\right)} \left[erfc\left(\frac{x_0 - U_{12}\frac{HBRn_i}{U_{22}} - \sqrt{HBR^2 - U_{22}^2\left(\frac{HBRn_i}{U_{22}}\right)^2}}{\sqrt{2}U_{11}} \right) \right.$$

$$+ erfc\left(\frac{x_0 + U_{12}\frac{HBRn_i}{U_{22}} - \sqrt{HBR^2 - U_{22}^2\left(\frac{HBRn_i}{U_{22}}\right)^2}}{\sqrt{2}U_{11}} \right)$$

$$- erfc\left(\frac{x_0 - U_{12}\frac{HBRn_i}{U_{22}} - \sqrt{HBR^2 - U_{22}^2\left(\frac{HBRn_i}{U_{22}}\right)^2}}{\sqrt{2}U_{11}} \right)$$

$$\left. - erfc\left(\frac{x_0 + U_{12}\frac{HBRn_i}{U_{22}} + \sqrt{HBR^2 - U_{22}^2\left(\frac{HBRn_i}{U_{22}}\right)^2}}{\sqrt{2}U_{11}} \right) \right]$$

Where the weighting and node values for a Chebyshev polynomial of the second kind of degree *N* for *i=1* to *N* can be characterized as:

$$n_i = \cos\left(\frac{i * \pi}{N+1}\right)$$

$$\omega_i = \frac{\pi}{(N+1)\sqrt{8\pi\left(1-n_i^2\right)}}\sin^2\left(\frac{i\pi}{N+1}\right)$$

Elrod[iii] found that a polynomial of degree 16 fit data with small relative errors on the order of 10^{-10} for a broad swath of example conjunctions. However, in more stressing cases found during unit testing this was determined to be insufficient. Given the low computational overhead of increasing the order, the default order for Pc calculation using the Elrod formulation is set to 64 but is configurable as an input to the routine.

2.1.4 2D Probability of Collision Elrod Function – Source Code Description

The primary function contained within the SDK used for estimating the 2D Probability of Collision of a close approach event using the Elrod formulation is the:

PcElrod.m

routine, which estimates the probability of collision using the formula above.

As inputs, the routine accepts the following:

Table 4: 2D Probability of Collision Elrod Routine Input Parameters

Input Variable	Definition
r1	[3X1] ECI Position Vector of the Primary Object (meters)
v1	[3X1] ECI Velocity Vector of the Primary Object (meters/second)
cov1	[6X6] Primary State covariance matrix corresponding to input primary object reference frame
r2	[3X1] ECI Position Vector of the Secondary Object (meters)
v2	[3X1] ECI Velocity Vector of the Secondary Object (meters/second)
cov2	[6X6] Secondary State covariance matrix corresponding to input primary object reference frame
HBR	Combined hard body radius or exclusion zone of the two objects (m)
Chebyshev_order	[Integer, Optional] Even Integer value for the order of the Chebyshev polynomial to be used to calculate the probability of collision, a higher order will return more accurate results, 16 is sufficient for all observed, short-duration encounters (Defaults to 16)
Warning_level	[Integer, Optional] Specifies warnings issued when encountering and remediating non-positive definite (NPD) 2x2 conjunction plane covariances (defaults to 3): 0 = No warnings issued when processing 2x2 covariances that are NPD. 1 = Warnings issued only for NPD covariances that cannot be remediated using the eigenvalue clipping methodwith the standard clipping factor of Fclip = 1e-4. 2 = Warnings issued for NPD covariances that cannot be remediated using Fclip = 1e-4, or for those that can be remediated but require a non-standard but acceptably small eigenvalue clipping value.

	3 = Warnings issued for all processed 2x2 NPD covariances.

The routine also accepts inputs in vectorized form and will give vectorized outputs; this is explained in greater detail in the code prologue itself.

The 2D Probability of Collision routine outputs the following:

Table 5: 2D Probability of Collision Elrod Routine Output Parameters

Output Variable	Definition
Pc	Probability of Collision calculated using Foster approximation in two dimensional space
Arem	[2X2] Combined covariance projected onto xz-plane in the relative encounter frame.
IsPosDef	Binary flag indicating if the combined, marginalized and remediated covariance has a negative eigenvalue. If the test fails, the Pc is not computed. The function returns NaN for Pc. (Success = 1 & Fail = 0)
IsRemediated	Binary Flag indicating if the combined and marginalized covariance was remediated

Validation cases for this algorithm are contained within the unit test suite for the SDK at:

..\TwoDimensionalPc\UnitTest\ProbabilityOfCollisionCode\PcElrod_UnitTest.m

These test cases were developed using previously defined stressing cases developed by Alfano 2009[ii], previously existing test cases developed by Omitron to test specific methods of cross-sectional area representation, and a test case designed to test error catching of non-positive definite covariance matrices.

Table 6: 2D Probability of Collision Elrod Function Unit Test Cases

Test ID	Description
test01	Alfano test case 1

test02	Alfano test case 2
test03	Alfano test case 3
test04	Alfano test case 4
test05	Alfano test case 5
test06	Alfano test case 6
test07	Alfano test case 7
test08	Alfano test case 8
test09	Alfano test case 9
test10	Alfano test case 10
test11	Alfano test case 11
test12	Omitron test case designed to test circular cross-sectional area calculation of probability of collision
test13	Omitron test case representing a real event where the covariance matrix was non-positive definite (algorithm remediates covariance and returns a Pc estimate of “0”)

2.2 Time of Close Approach Correction

Occasionally on delivered CDMs, the TCA of an event may be misreported due to several possible factors. The reported TCA may be at the beginning or end of a screening span, or the conjunction may be of significantly long encounter duration and with low relative velocity such that TCA determination is more difficult. In most cases the reported TCA is off by only a small amount and often is off by less than the TCA reporting tolerance (0.0005 s).

While rare, several cases of large corrections to the reported TCA have been observed in operational data; these cases were then used as unit test criterion.

2.2.1 Time of Close Approach Correction – Mathematical Formulas

There are two readily apparent methods by which to determine the TCA correction for a given CDM. The first is through a linear trajectory approximation. The second is through two-body propagation and iteration to a convergence criteria. Only the linear trajectory approximation is currently implemented.

The correction to a TCA using a linear trajectory approximation is easily accomplished through the knowledge that for two vectors, the closest point between these two vectors occurs when the dot product of the relative position and velocity of these vectors is equal to zero:

$$\vec{r}_{rel} \bullet \vec{v}_{rel} = 0$$

For a linear approximation, the relative velocity vector is constant and the relative position vector is a function of the initial relative position estimate, the relative velocity vector, and time since the initial TCA estimate so the correction to TCA may be determined by:

$$\delta_{TCA} = -\frac{\vec{r}_{rel} \bullet \vec{v}_{rel}}{\|\vec{v}_{rel}\|^2}$$

If only performing the linear trajectory approximation, this is the final step in estimating the correction to TCA, for additional accuracy this can be iterated upon using two-body propagation. In this case, the initial correction estimate is used to propagate the initial states to this new TCA estimate using two-body propagation, then iterating upon this process until the TCA correction meets some convergence criteria.

2.2.2 Time of Close Approach Correction – Source Code Description

The primary function contained within the SDK used for assessing the time of close approach correction is the:

FindNearbyCA.m

routine, which assesses any required change in the estimation of the TCA.

As inputs, the routine accepts the following:

Table 7: TCA Correction Routine Input Parameters

Input Variable	Definition
X1	Primary object's pos/vel state vector in ECI coordinates, (6x1) [m & m/s] or [km & km/s]
X2	Secondary object's pos/vel state vector in ECI coordinates, (6x1) [m & m/s] or [km & km/s]
MotionMode	Optional input to denote which method use for TCA correction approximation. Allowable values: - 'LINEAR' (Currently Implemented) - 'TWOBODY' (Not Currently Implemented)
RelTol	Tolerance level for determining TCA correction when using 'TWOBODY' motion mode. (Not currently Used)

The TCA Correction routine outputs the following:

Table 8: TCA Correction Routine Output Parameters

Output Variable	Definition
dTCA	Offset time to TCA from initial assessment (s)
X1CA	Corrected primary ECI state at CA (6x1) [m & m/s] or [km & km/s]
X2CA	Corrected secondary ECI state at CA (6x1) [m & m/s] or [km & km/s]

Validation cases for this algorithm are contained within the unit test suite for the SDK at:

..\TwoDimensionalPc\UnitTest\ProbabilityOfCollisionCode\FindNearbyCA_UnitTest.m

Table 9: TCA Correction Function Unit Test Cases

Test ID	Description
test01	Operational event with a dTCA of -497.068 s
test02	Operational event with a dTCA of 28.329 s
test03	Operational event with a dTCA of 27.198
test04	Operational event with a dTCA of -0.173 s

2.3 Covariance Remediation

Hall et. al.[v] previously examined the occurrence rate of non-positive definite covariance matrices being delivered within CDMs and found that while infrequent, these events did occur. This posed a more frequent risk in the past before CDMs shifted to reporting a greater number of significant figures in the covariance entries on the CDM, but does still occasionally crop up in more recently delivered CDMs. Non-positive definite covariance matrices can prove problematic, as Pc calculations require that the combined state covariance matrix of the two objects be positive definite.

2.3.1 Covariance Remediation – Mathematical Formulas

To alleviate these potential issues, a method of forcing the combined covariance matrix in the 2 element conjunction plane to be positive definite was proposed through the use of Eigenvalue clipping. i.e. forcing the Eigenvalues of an input matrix to be greater than some clipping value and then reforming the covariance matrix using the remediated Eigenvalues and Eigenvectors. Reducing the covariance matrix to the 2X2 conjunction plane greatly reduces the frequency at which non-positive definite covariance matrices arise causing a breakdown in the probability of collision formulation.

To begin with, the covariance matrices of the primary and secondary objects are combined in a common reference frame and reduced to a 2X2 covariance matrix (A) in the 2D conjunction plane (see Section 2.1.1) then the Eigenvalues (λ) and Eigenvectors (V) of this covariance matrix are determined.

$$A = V\lambda_{raw}V'$$

If any of the Eigenvalues (λ_i) is less than a prescribed clipping value (Hall et. al.[v] recommends using a value equal to (**1.0E-4*HBR)^2**), that Eigenvalue is replaced by the clipping value (λ_{clip}) resulting in a new set of Eigenvalues (λ_{rem}).

$$\lambda_{raw} < \lambda_{clip} = \lambda_{clip} \xrightarrow{yields} \lambda_{rem}$$

A new, remediated covariance is then formed using these new Eigenvalues and used for probability of collision assessment.

$$A_{rem} = V\lambda_{rem}V'$$

2.3.2 Covariance Remediation – Source Code Description

The primary function contained within the SDK used for remediating non-positive definite covariance matrices in the 2X2 conjunction plane is the:

CovRemEigValClip.m

routine, which assesses the positive definite disposition of the combined covariance matrix and remediates that covariance if needed.

As inputs, the routine accepts the following:

Table 10: Covariance Remediation Routine Input Parameters

Input Variable	Definition
Araw	Input 2X2 combined covariance matrix in the conjunction plane.
Lclip	Clipping limit for eigen values, optional, defaults to 0 m. Recommended as: $(1.0E\text{-}4 * HBR)^2$
Lraw	Eigenvalues of Araw [2x1], optional, will be calculated within code if not input
Vraw	Eigenvector matrix of Araw [2x2], optional, will be calculated within code if not input

The Covariance Remediation routine outputs the following:

Table 11: Covariance Remediation Routine Output Parameters

Output Variable	Definition
Lrem	Post-remediation eigen values [2X1]
Lraw	Pre-remediation eigen values [2X1]
Vraw	Eigenvector matrix [2X2]
PosDefStatus	Positive Definite Status of Araw -1 => Araw is non-positive definite 0 => Araw is positive semi-definite 1 => Araw is positive definite
Clip Status	Boolean flag indicating if clipping was required. True = clipping required
Adet	Determinant of remediated covariance [1X1]
Ainv	Inverse of remediated covariance matrix [2X2]

Arem	Remediated covariance matrix [2X2]

Validation cases for this algorithm are contained within higher level unit test suites for the SDK within:

..\TwoDimensionalPc\UnitTest\ProbabilityOfCollisionCode\Pc2D_Foster_UnitTest.m

..\TwoDimensionalPc\UnitTest\ProbabilityOfCollisionCode\PcElrod_UnitTest.m

Table 12: TCA Correction Function Unit Test Cases

Test ID	Description
test15	Omitron test case representing a real event where the covariance matrix was non-positive definite (algorithm remediates covariance and returns a Pc estimate of “0”)

2.4 Coppola Conjunction Bounds

Coppola[vi] developed a formula to characterize the encounter duration for the conjunction of two objects and introduced the concept of a short-term encounter validity interval that characterizes the total encounter time under which the short-term assumptions are assumed met. This metric provides the means for assessing whether a conjunction satisfies the short encounter assumption so that the standard collision probability metric is valid.

2.4.1 Coppola Conjunction Bounds – Mathematical Formulas

The conjunction bounds as defined by Coppola stem from the same assumptions made as part of the 2D Pc calculation as defined in Section 2.1. Namely, that the encounter occurs over so small a time interval that the motion of the objects can be assumed to be linear (i.e., straight lines). Second, the velocity uncertainty is assumed to be sufficiently small that it can be treated as zero and ignored.

To determine the conjunction bounds the encounter is mapped to an "encounter" reference frame similar to the process used for calculation of 2D Pc.

$$\hat{x} = \frac{\vec{v}_{rel}}{\|\vec{v}_{rel}\|}$$

$$\hat{y} = \vec{r}_{rel} - \hat{x} * (\hat{x}^T \vec{r}_{rel})$$

$$\hat{z} = \hat{x} \times \hat{y}$$

The relative position vector and 3X3 combined position covariance matrix are then rotated to this frame:

$$ROT_{xyz} = \begin{bmatrix} \hat{x}^T \\ \hat{y}^T \\ \hat{z}^T \end{bmatrix}$$

$$\vec{r}_{xyz} = ROT_{xyz} \vec{r}_{rel}$$

$$P_{xyz} = ROT_{xyz} P_{eci} ROT_{xyz}^T$$

From this transformation the following quantities are defined:

$$P_{xyz} = \begin{bmatrix} \eta^2 & w^T \\ w & Pc \end{bmatrix}$$

$$b = (w^T Pc^{-1})^T$$

$$\sigma_v^2 = \eta^2 - b^T w$$

$$q_0 = b^T \vec{r}_{xyz}(2:3)$$

The initial and final times spanning the encounter region are then characterized by an inverse error function and the above quantities where γ defines the resolution of the approximation. For machine precision this should be set to 1E-16, but may be set to other values:

$$\alpha_c = erfc^{-1}(\gamma)$$

$$\tau_0(\gamma) = \frac{-\sqrt{2}\alpha_c(\gamma)\sigma_v + q_0 - HBR\sqrt{1 + b^T b}}{\vec{v}_{rel}}$$

$$\tau_1(\gamma) = \frac{\sqrt{2}\alpha_c(\gamma)\sigma_v + q_0 + HBR\sqrt{b^T b}}{\vec{v}_{rel}}$$

2.4.2 Coppola Conjunction Bounds – Source Code Description

The primary function contained within the SDK used for determining the Coppola linear conjunction bounds is the:

LinearConjDuration.m

routine, which assesses the encounter duration for a given input conjunction.

As inputs, the routine accepts the following:

Table 13: Coppola Conjunction Bounds Routine Input Parameters

Input Variable	Definition
r1	Primary object's position vector in inertial coordinates [1x3 or 3x1] (m or km)
v1	Primary object's velocity vector in inertial coordinates [1x3 or 3x1] (m or km)
cov1	Primary object's inertial (r,v) state covariance matrix [3x3 or 6x6] (in units consistent with r1 & v1)
r2	Secondary object's position vector in inertial coordinates [1x3 or 3x1] (m or km)
v2	Secondary object's velocity vector in inertial coordinates [1x3 or 3x1] (m or km)
cov2	Secondary object's inertial (r,v) state covariance matrix [3x3 or 6x6] (in units consistent with r1 & v1)
HBR	Hard body radius (in same units as r1 and r2)
params	Optional run parameters

	- params.gamma = The precision factor for the Coppola duration analysis (default = 1e-16) [1x1] (dimensionless) - params.FindCA = Flag to refine CA point before the analysis (default = false) - params.verbose - Flag for verbose operation, used for development and debugging (default = false).

The Coppola conjunction bounds routine outputs the following:

Table 14: Coppola Conjunction Bounds Routine Output Parameters

Output Variable	Definition
tau0	Initial time bound for the conjunction relative to TCA
tau1	Final time bound for the conjunction relative to TCA
dtau	Conjunction duration = tau1-tau0, which spans the time that the collision probability grows from zero to its final value (to within the precision factor gamma), over the period TCA+tau0 <= time <= TCA+tau1
taum	Midpoint time for the conjunction relative to TCA taum = (tau1+tau0)/2 which is approximately the time that the probability rate peaks for a linear conjunction.
delt	STEVI half-width, measuring the time before and after TCA TCA-delt <= time <= TCA+delt that the linear-trajectory and constant-covariance assumptions must hold

Validation cases for this algorithm are contained within the SDK in the following location:

..\TwoDimensionalPc\UnitTest\ProbabilityOfCollisionCode\LinearConjDuration_UnitTest.m

Table 15: Coppola Conjunction Bounds Routine Unit Test Cases

Test ID	Description
test01	Based on the conjunction plotted in Figure 1 of Hall et al. AAS 18-244
test02	Based on the conjunction plotted in Figure 4 of Hall et al. AAS 18-244

2.5 2D Probability of Collision Usage Boundaries

The two-dimensional (2D) probability of collision (Pc) estimation method relies on several assumptions that must be satisfied for accurate results. Previous analysis by Hall[vii] found that while for most operationally experienced conjunctions the 2D Pc is sufficient, there are rare cases where the 2D Pc algorithms significantly underestimate the true Pc as calculated through Monte Carlo analysis this occurred in roughly 0.05% of operational cases with a reported 2D Pc greater than 1.00E-7. To this end, a series of usage boundary tests were proposed to identify events with suspect 2D Pc calculations and report these violations. If a violation is reported, it is generally recommended to re-examine the Pc by running a Monte Carlo analysis of the conjunction.

2.5.1 2D Probability of Collision Usage Boundaries – Mathematical Formulas

A series of 4 tests as recommended by Hall[vii] are implemented within the Pc Omnibus tool to identify when 2D Pc estimation is suspect and report these violations to the user. These tests are as follows:

1. Conjunction duration boundaries
2. Equinoctial covariance NPD effect boundaries
3. Offset from TCA 2D Pc variation amplitude boundaries
4. Offset from TCA 2D Pc number of extrema boundaries

The conjunction duration boundaries are based on the relationship between the short-term encounter validity interval (STEVI) and the smaller of the orbital periods of the two objects in question on a given CDM. Essentially this test throws a violation if the period of time which two objects spend in proximity to each other is overly long. The STEVI is based on a maximum span based on the conjunction duration calculations described in Section 2.4.

$$STEVI = \max(\tau_1 - \tau_0, |\tau_0|, |\tau_1|)$$

$$R = \frac{STEVI}{Period}$$

The equinoctial covariance NPD effect boundaries are based on the change induced in the 2D Pc calculation due to covariance remediation in equinoctial element space. As mentioned in Section 2.3, occasionally reported covariance matrices are NPD, and these covariance matrices can be remediated through a process known as Eigenvalue clipping. In Cartesian coordinates, the NPD covariance matrices often arise due to scale differences between the position and velocity term Eigenvalues. To alleviate this, the effect of covariance remediation is examined in the equinoctial frame. The object states and covariance matrices are mapped to the equinoctial frame, and the equinoctial covariance matrices are remediated if NPD. These remediated covariance matrices are then mapped back to the Cartesian frame, and the Pc is recalculated. The ratio between the Pc values calculated using the remediated and nominal covariance matrices is then examined to assert violations of the boundary conditions.

$$A = \log_{10} \frac{Pc_{Rem}}{Pc_{Nom}}$$

The offset from TCA 2D Pc variation amplitude boundaries examines the differences between the maximum Pc at any point during the encounter interval and the Pc at the midpoint of the encounter interval. For most operational events, the offset time can span the entire short-term encounter validity interval and still yield approximately the same 2D-Pc value. Some conjunctions, however, show large offset-from-TCA 2D-Pc variations over their short-term encounter validity intervals. An example of both of these cases can be observed in Figure 1.

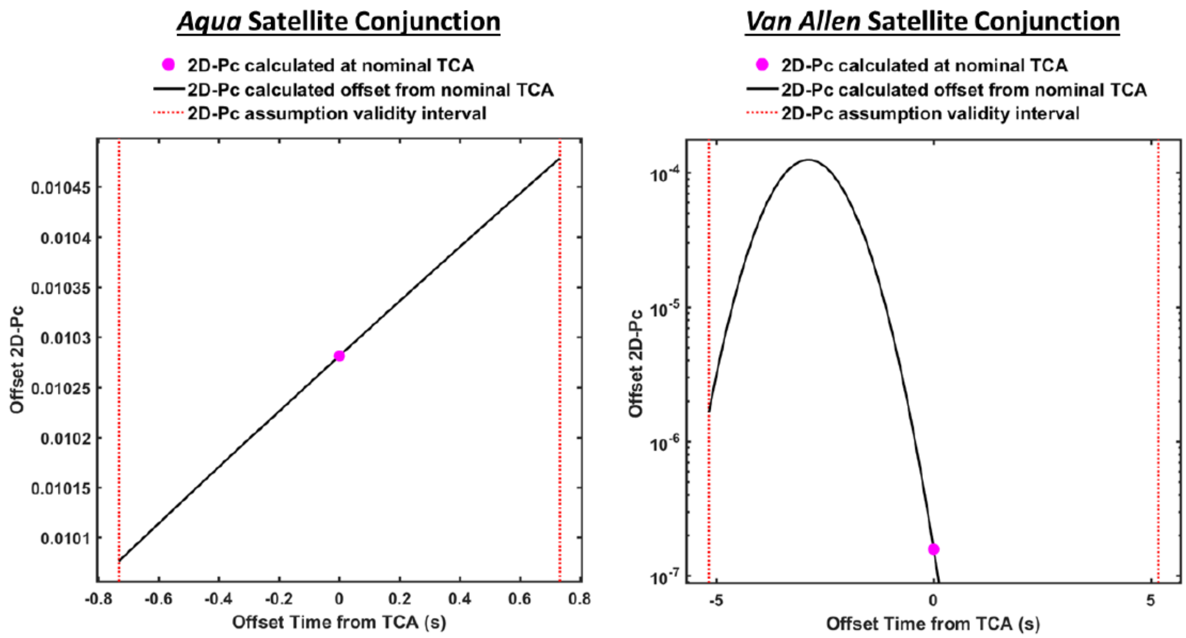

Figure 1: Offset-From-TCA 2D Pc Estimates for an *Aqua* Conjunction (left) for which 2D Pc shows little variation over the encounter interval, and a *Van Allen* Conjunction for which 2D Pc shows significant variation over the encounter interval

The variation amplitude is calculated by the following relationship between the maximum Pc over the encounter interval and the Pc calculated at the midpoint of the encounter interval:

$$V = \log_{10} \frac{Pc_{max}}{Pc_{Mid}}$$

The offset from TCA 2D Pc number of extrema boundaries examines the number of extrema in the offset from TCA 2D Pc variation curve over the encounter interval, values greater than 1 may indicate a repeating conjunction.

$$N = count(Minima + Maxima)$$

These evaluations are then compared to the boundary usage conditions defined in the Table 16:

Table 16: 2D Pc Usage Violation Levels and Boundaries

Usage Criterion and Boundary Metric	Level 0 None or Very-Low	Level 1 Low	Level 2 Medium	Level 3 High	Level 4 Very-High
Conjunction duration (short-term encounter validity interval) R = STEVI/Min.Period	$\frac{1}{36} < R \leq \frac{1}{27}$	$\frac{1}{27} < R \leq \frac{1}{18}$	$\frac{1}{18} < R \leq \frac{1}{9}$	$\frac{1}{9} < R \leq \frac{1}{3}$	$R > \frac{1}{3}$
TCA equinoctial covariance NPD remediation amplitude A = |Log(PcRem/PcRaw)|	$0 \leq A \leq .003$	$.003 < A \leq .01$	$.01 < A \leq 0.03$	$.03 < A \leq 0.1$	$A > 0.1$
Offset-from-TCA 2D-Pc curve over STEVI variation amplitude metric[2] V = Log_{10}(PcMax/PcMid)	$V \leq 0.41$	$.41 < V \leq 0.55$	$.55 < V \leq 0.80$	$.80 < V \leq 1.30$	$V > 1.30$
Offset-from-TCA 2D-Pc curve over STEVI number of extrema N = #(minima+maxima)	$N \leq 1$	$N = 2$	$N = 3$	$N = 4$	$N \geq 5$

Violations are then characterized numerically by severity.

2.5.2 2D Probability of Collision Usage Boundaries – Source Code Description

The primary function contained within the SDK used for assessing 2D Pc usage boundary condition violations is the:

Pc2DUsageBoundaries.m

routine, which assesses each of the four 2D Pc usage criteria and their respective boundary conditions.

As inputs, the routine accepts the following:

Table 17: 2D Probability of Collision Usage Boundaries Routine Input Parameters

Input Variable	Definition
X1TCA	Primary Cartesian TCA state (m) [6x1]
C1TCA	Primary Cartesian TCA covariance (m^2 & $(m/s)^2$) [6x6]
X2TCA	Secondary Cartesian TCA state (m) [6x1]
C2TCA	Secondary Cartesian TCA covariance (m^2 & $(m/s)^2$) [6x6]

HBR	Combined hard body radius (m)

The 2D probability of collision usage boundaries routine outputs the following:

Table 18: 2D Probability of Collision Usage Boundaries Routine Output Parameters

Output Variable	Definition
UsageLevel	Combined 2D-Pc method usage boundary level [1x1 double array holding an integer]
UsageLevels	Individual 2d-Pc method usage boundary levels for the four types (A,B,C,D) of 2D-Pc method usage boundaries analyzed: A) Conjunction Duration Usage Boundaries B) Equinoctial Covariance NPD Usage Boundaries C) Offset-from-TCA Variation Usage Boundaries D) Offset-from-TCA Number of Extrema Usage Boundaries [1x4 double array holding integers]
UsageMessages	The 2D-Pc method usage messages for the four types of usage boundaries (A,B,C,D) analyzed [1x4 cell array]
UsageInfo	Structure with detailed information on the 2D-Pc usage boundary evaluation analysis [structure]

Validation cases for this algorithm are contained within the following unit test suites for the SDK within:

..\TwoDimensionalPc\UnitTest\ProbabilityOfCollisionCode\Pc2DUsageBoundaries_UnitTest.m

Table 19: 2D Probability of Collision Usage Boundaries Routine Unit Test Cases

Test ID	Description

test01	Based on the event plotted in Fig.1 of Hall et al (2018) AAS 18-244 evaluated with no 2D-Pc method usage boundary violations.
test02	Based on the event plotted in Fig.4 of Hall et al (2018) AAS 18-244b evaluated with an overall VERY-HIGH level of 2D-Pc method usage boundary violation.
test03	Based on an event with at-TCA equinoctial covariance NPD issues evaluated with an overall VERY-HIGH level of 2D-Pc method usage boundary violation.
test04	Based on an interaction between two closely spaced LEO objects, with an extended STEVI conjunction duration, leading to multiple close approaches (i.e., a repeating conjunction), and evaluated with an overall VERY-HIGH level of 2D-Pc usage boundary violation.

2.6 Maximum 2D Probability of Collision Calculation

In the conjunction assessment community, there is sustained interest in determining not only the current probability of collision, but also the maximum probability of collision. This is due to the probability of collision having a possibility of producing a false sense of security for occasions when the conjunction is not truly well characterized. In this section two methods are examined for determining the maximum probability of collision:

First, when the orbital position uncertainties are high, the reported probability of collision may be low due to the sheer dilution of the combined covariance matrices. In this case the Pc is referred to as diluted and may understate the collision risk.

Second, if covariance data is unavailable for a specific object, no assumptions may be made about the object's covariance, so there is an alternative measure of maximum probability of collision based on having no knowledge of one of the object's covariance matrix.

2.6.1 Dilution Region Assessment of Maximum Probability of Collision – Mathematical Formulas

Hejduk[viii] outlines the method used to determine the dilution status and relevant maximum Pc when a conjunction is in the dilution region and builds on earlier bodies of work. The dilution effect can be inferred from Figure 2 where for a given HBR, there will be a particular joint covariance size that will maximize the amount of covariance probability density that falls within that HBR and thus will similarly maximize the calculated Pc. Because such a Pc maximum exists, growing or shrinking the covariance from this value will produce smaller Pc values.

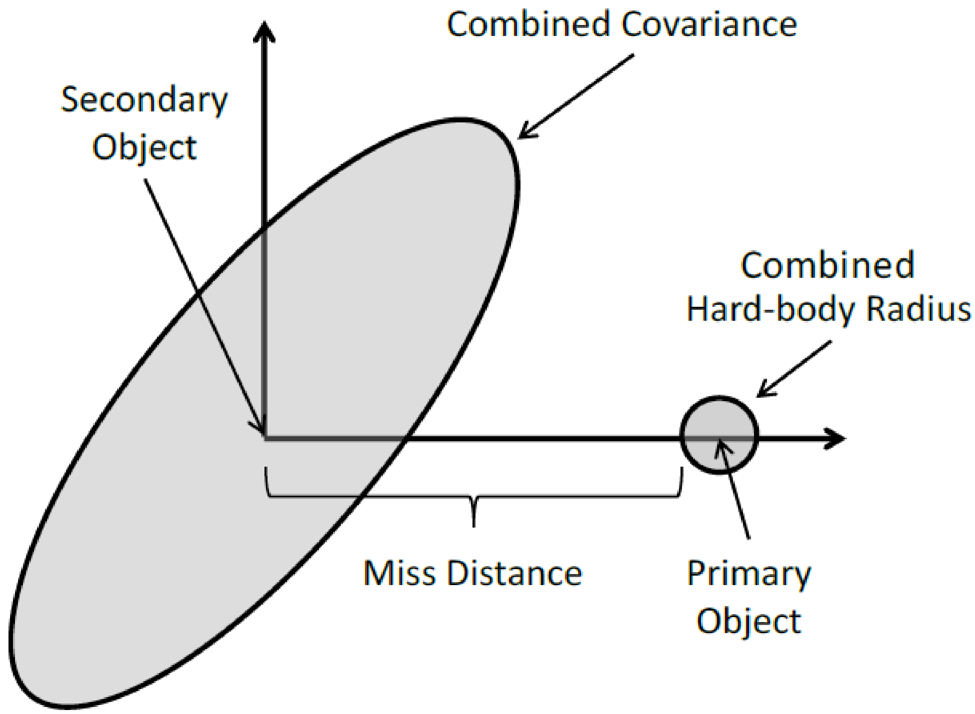

Figure 2: Conjunction-Plane rendering of 2-D Pc Calculation

This can also be observed in Figure 3, which demonstrates that as a covariance matrix expands, the Pc either grows in the robust region, or decreases in the dilution region. To assess for a given CDM whether a conjunction is in the dilution region or not, either the primary covariance, the secondary covariance, or the combined covariance matrix is scaled and assessed vs the initial Pc estimate. By scaling the covariance matrices individually, each object can be assessed for whether or not it is in the dilution region by examining the first derivative of Pc with respect to a linear scaling factor of one of the input covariance matrices.

$$\frac{\partial Pc}{\partial C_{pri}} > 1 \xrightarrow{yields} Primary\ Object\ in\ Robust\ Region$$
$$\frac{\partial Pc}{\partial C_{pri}} < 1 \xrightarrow{yields} Primary\ Object\ in\ Dilution\ Region$$
$$\frac{\partial Pc}{\partial C_{sec}} > 1 \xrightarrow{yields} Secondary\ Object\ in\ Robust\ Region$$
$$\frac{\partial Pc}{\partial C_{sec}} < 1 \xrightarrow{yields} Secondary\ Object\ in\ Dilution\ Region$$

When these first derivatives are equal to 0, the maximum Pc with regards to the input object dilution has been reached. This is determined via an iterative process until convergence is attained by examining a span of scaled covariance matrices and refining this span until the maximum Pc is determined. As covariance matrices do not generally grow with data updates, if an object is in the robust region, the maximum Pc is reported as the Pc for the input CDM.

$$Pc_{max} = max\begin{pmatrix} Pc_{input} \\ Pc_{max}(Primary\ Dilution) \\ Pc_{max}(Secondary\ Dilution) \end{pmatrix}$$

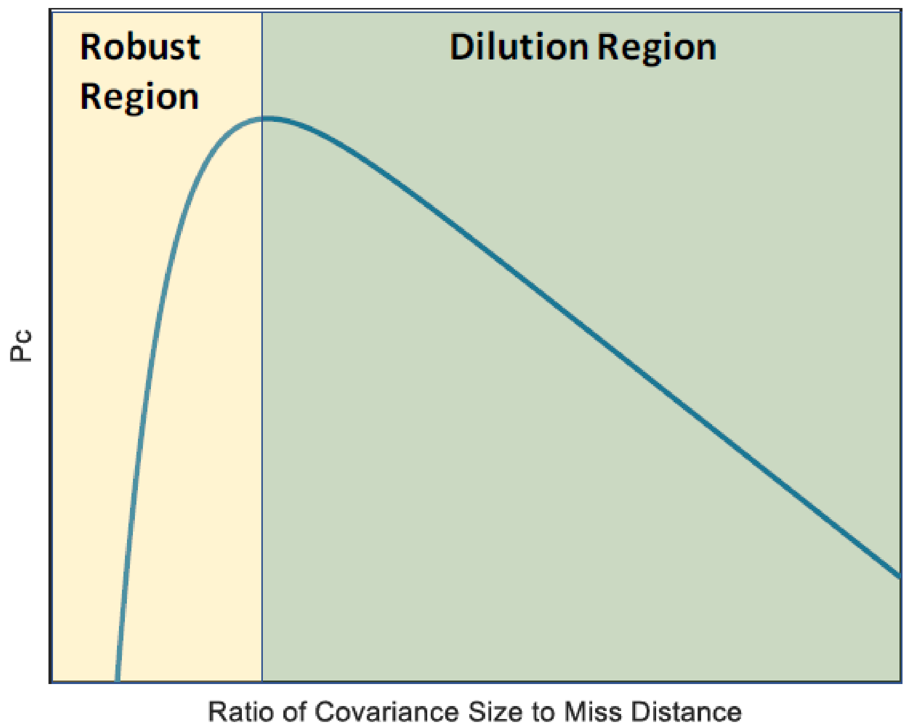

Figure 3: Pc vs the Ratio of Covariance Size to Miss Distance

2.6.2 Dilution Region Assessment of Maximum Probability of Collision – Source Code Description

The primary function contained within the SDK used for estimating the maximum 2D Probability of Collision of a close approach event with regards to dilution region assessment is the:

DilutionMaxPc.m

routine, which estimates the maximum probability of collision using the process above.

As inputs, the routine accepts the following:

Table 20: Dilution Region Maximum Probability of Collision Routine Input Parameters

Input Variable	Definition

r1	[3X1] ECI Position Vector of the Primary Object (meters)
v1	[3X1] ECI Velocity Vector of the Primary Object (meters/second)
cov1	[6X6] Primary State covariance matrix corresponding to input primary object reference frame
r2	[3X1] ECI Position Vector of the Secondary Object (meters)
v2	[3X1] ECI Velocity Vector of the Secondary Object (meters/second)
cov2	[6X6] Secondary State covariance matrix corresponding to input primary object reference frame
HBR	Combined hard body radius or exclusion zone of the two objects (m)
params	Run parameters for subfunction “PcDilution.m” (optional)

The maximum 2D Probability of Collision routine outputs the following:

Table 21: Dilution Region Maximum Probability of Collision Routine Output Parameters

Output Variable	Definition
PcMax	Maximum probability of collision value from combined primary and secondary object covariance scaling Pc-dilution analysis
Diluted	Integer indicating if the either the primary or secondary object is in the dilution region: Diluted = 0 => No dilution for either case Diluted = 1 => Secondary dilution but no primary dilution Diluted = 10 => Primary dilution but no secondary dilution Diluted = 11 => Primary dilution and secondary dilution
Pri	Structure holding the primary Pc-dilution analysis results
Sec	Structure holding the secondary Pc-dilution analysis results

Validation cases for this algorithm are contained within the unit test suite for the SDK at:

..\TwoDimensionalPc\UnitTest\ProbabilityOfCollisionCode\DilutionMaxPc _UnitTest.m

These test cases were developed using previously existing test cases developed by Omitron to test specific stressing cases observed operationally.

Table 22: 2D Probability of Collision Foster Function Unit Test Cases

Test ID	Description
test01	Operational close approach event with maximum estimated probability of collision from selected subset of events using original hard body radius of 20 meters.
test02	Operational close approach event with maximum secondary object radial position uncertainty from selected subset of events using a modified hard body radius of 100 meters for more rapid testing.
test03	Operational close approach event with maximum secondary object intrack position uncertainty from selected subset of events using a modified hard body radius of 100 meters for more rapid testing.
test05	Operational close approach event with minimum miss distance from selected subset of events using a modified hard body radius of 20 meters for more rapid testing.
test06	Operational close approach event with minimum relative velocity from selected subset of events using original hard body radius of 20 meters.

2.6.3 Frisbee's Method of Determining Maximum 2D Probability of Collision – Mathematical Formulas

Frisbee 2015[ix] proposed a method by which the maximum possible probability of collision could be determined for a close approach event for which only one object has position uncertainty information. This is of particular use in determining whether an encounter may be of risk to an asset, as the maximum probability of collision may be below an actionable threshold. To determine the maximum probability of collision, the covariance ellipsoid of the object possessing a covariance matrix is mapped to the conjunction plane and distended so that the Mahalanobis distance between the two objects is equal to one. To do this, the covariance of the secondary object is oriented along a one dimensional position uncertainty along the miss vector between the two objects. Graphically, this can be seen in Figure 4:

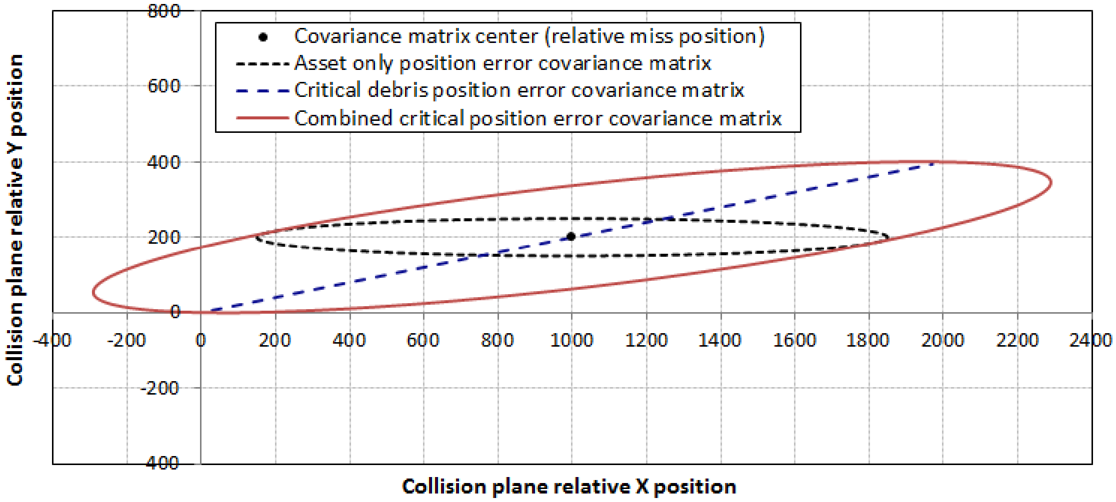

Figure 4: Graphic Representation on Maximum Pc Covariance in the Conjunction Plane[ix]

Since this unknown uncertainty resulting in maximum probability of collision is oriented along the miss vector, it may be characterized using a constant in conjunction with the relative miss vector, $\boldsymbol{u_{rel}}$, which is the unit vector of the miss geometry in the conjunction plane.

$$\boldsymbol{C_{unknown}} = \boldsymbol{V_r u_{rel} u_{rel}^T}$$

From Section 2.1, we know that the probability of collision may be characterized as:

$$P_C = \frac{1}{2\pi\sqrt{|\boldsymbol{C}|}} \int_{\oplus} e^{-\frac{1}{2}(\vec{r}-\vec{r}_d)^T \boldsymbol{C}^{-1}(\vec{r}-\vec{r}_d)}\, d\vec{r}$$

Where $\boldsymbol{C}$ is the combined position uncertainty of the two objects for probability of collision calculation, and is characterized as the sum of the two position covariances in a common frame.

$$\boldsymbol{C} = \boldsymbol{C_{known}} + \boldsymbol{C_{unknown}}$$

Frisbee 2015[ix], makes an argument for an approximation of the probability of collision calculation assuming that the spatial debris density does not vary over the exclusion zone reducing the two dimensional probability of collision equation to:

$$P_C \approx A \frac{e^{-0.5 r_{rel}^T C^{-1} r_{rel}}}{2\pi\sqrt{C}}$$

By differentiating this equation with respect to V_r, it is possible to determine the value of V_r which maximizes the probability of collision with respect to the known object position uncertainty and miss vector.

$$V_r = |r_{rel}|^2 \left(\frac{K_A^2 - 1}{K_A^2}\right)$$

Where:

$$K_A^2 = r_{rel}^T \boldsymbol{C}_{\boldsymbol{known}}^{-\boldsymbol{1}} r_{rel}$$

As stated before, the Mahalanobis distance of the miss geometry becomes a value of 1, this causes Frisbee's approximation to reduce to:

$$P_C \approx A \frac{e^{-0.5}}{2\pi\sqrt{C}}$$

Frisbee's approximation functions simplify the calculation of the maximum probability of collision by removing the integration of the debris spatial density from the probability of collision equation instead opting to multiply the debris spatial density at the time of closest approach by the area of the exclusion zone. This is effectively making the assumption that the debris spatial density is constant over the entire cross-sectional area of the exclusion zone. This causes the approximation to tend to overestimate the actual probability of collision, and can be seen in Figure 5 where the example event from Frisbee's paper is analyzed for a varying array of hard body radii.

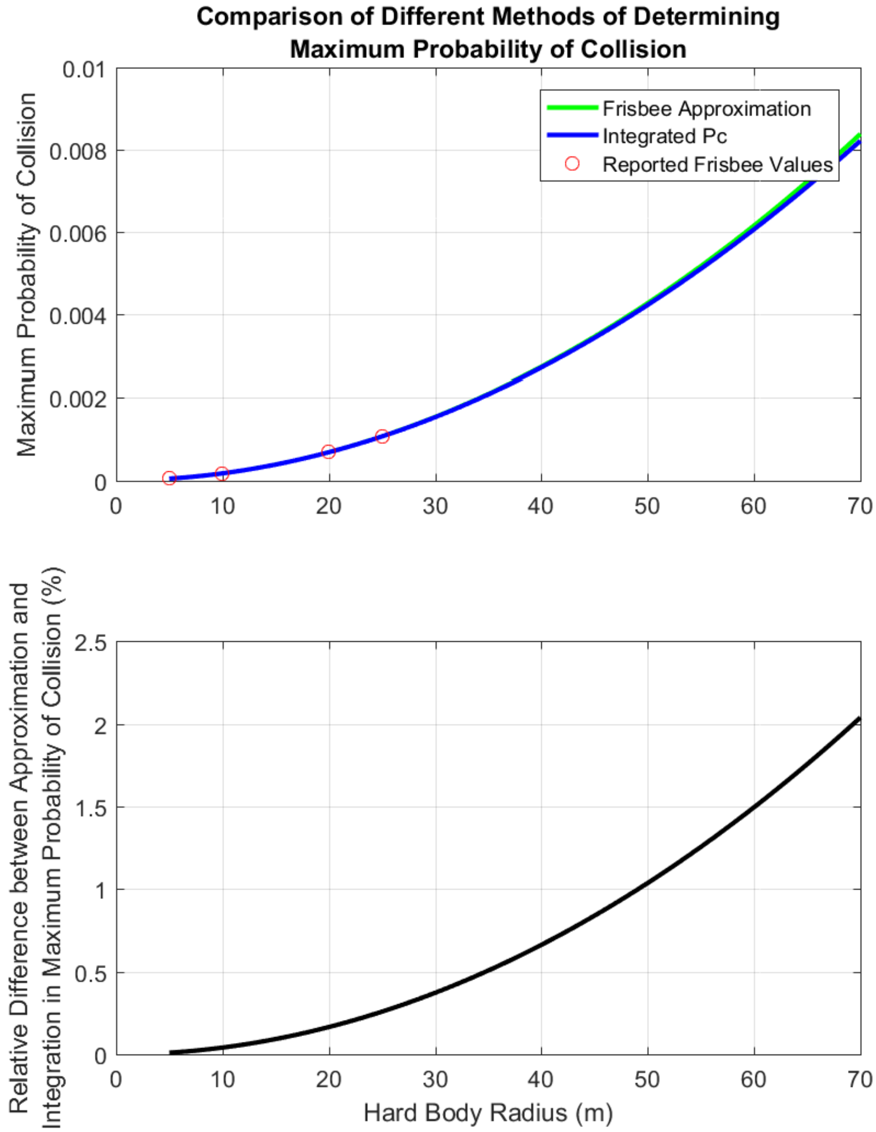

Figure 5: Comparison of Frisbee's Approximation to the Integrated Maximum Probability of Collision for Frisbee's Example.

In Frisbee's manufactured example close approach, the difference between the maximum probability of collision approximation and the full integration differs by only a few percent even as the input hard body radius approaches an upper limit defined by the ISS station size.

While the assumption that the debris spatial density is invariant over the exclusion zone is a valid one that causes only minor variations when the exclusion radius is small in respect to the combined position uncertainty bounds of the event, this assumption becomes less valid as the exclusion zone increases in size. Particularly with respect to the combined position uncertainty bounds. In Figure 6, an extreme operational example is analyzed with varying hard body radii, and Frisbee's approximation begins to give answers that no longer make physical sense as the approximated probability of collision exceeds a value of unity. For this reason, Omitron has coded its output probability of collision to reflect the debris spatial density as integrated over the entire exclusion zone instead of using the approximation.

This will give operators a better measure of the maximum probability of collision and the outputs will be more robust in that they will not give non-sensical results under specific conditions.

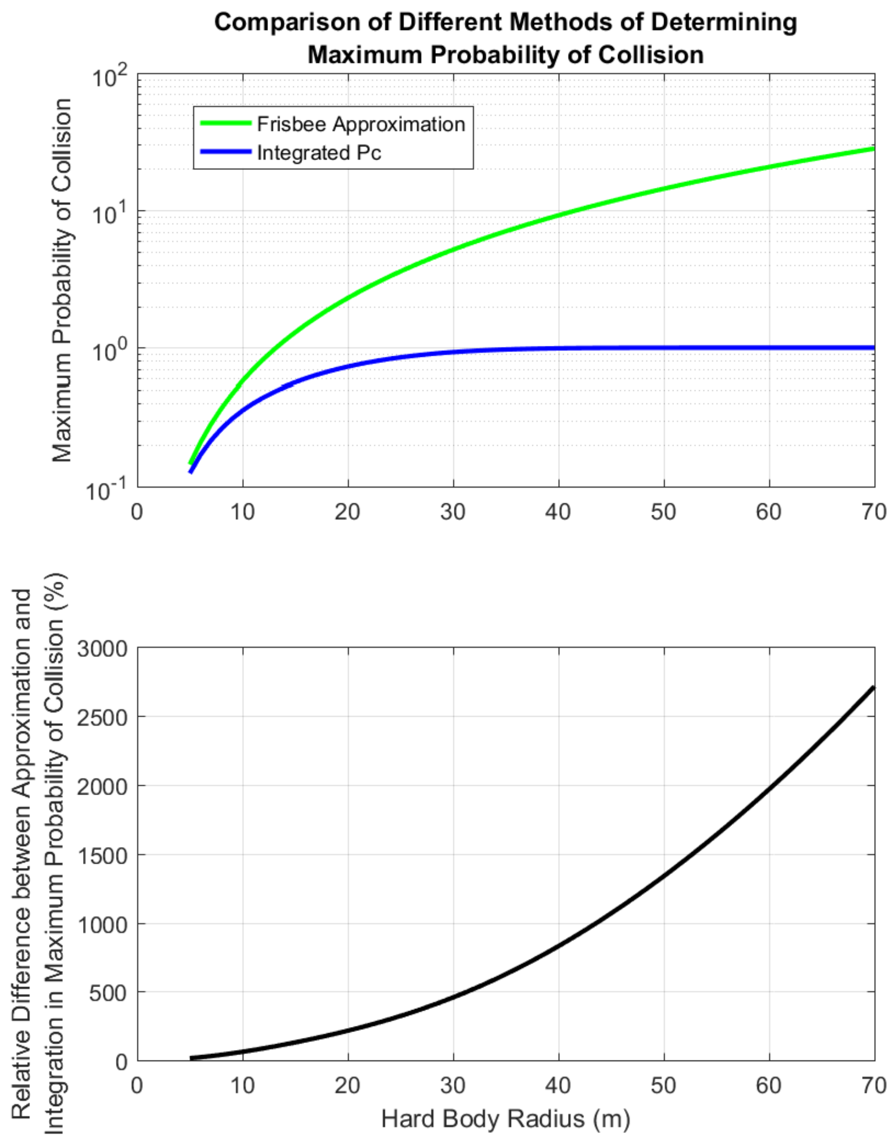

Figure 6: Comparison of Frisbee's Approximation to the Integrated Maximum Probability of Collision for Collision with a Pc = 4.20E-01.

2.6.4 Frisbee's Method of Determining Maximum 2D Probability of Collision – Source Code Description

The primary function contained within the SDK used for estimating the maximum 2D Probability of Collision of a close approach event is the:

FrisbeeMaxPc.m

routine, which estimates the probability of collision using the formula above.

As inputs, the routine accepts the following:

Table 23: Maximum 2D Probability of Collision Routine Input Parameters

Input Variable	Definition
r1	[3X1] ECI Position Vector of the Primary Object (meters)
v1	[3X1] ECI Velocity Vector of the Primary Object (meters/second)
cov1	[6X6] Primary State covariance matrix corresponding to input primary object reference frame
r2	[3X1] ECI Position Vector of the Secondary Object (meters)
v2	[3X1] ECI Velocity Vector of the Secondary Object (meters/second)
cov2	[6X6] Secondary State covariance matrix corresponding to input primary object reference frame
HBR	Combined hard body radius or exclusion zone of the two objects (m)
RelTol	Relative Tolerance used for double integration convergence (1E-08 is recommended)
HBRType	Definition of hard body region, typically "circle". Allowable inputs: • "circle" – Hard body region defined as a sphere or circle • "square" – Hard body region defined as a cube or square

	• "squareEquArea" – Hard body region defined as a square with equivalent area to a circle with radius as defined y HBR

The Maximum 2D Probability of Collision routine outputs the following:

Table 24: Maximum 2D Probability of Collision Routine Output Parameters

Output Variable	Definition
Pc	Maximum Probability of Collision calculated using Frisbee's Method

Validation cases for this algorithm are contained within the unit test suite for the SDK at:

..\TwoDimensionalPc\UnitTest\ProbabilityOfCollisionCode\FrisbeeMaxPc_UnitTest.m

These test cases were developed using previously defined stressing cases developed by Alfano 2009[ii], manufactured test cases corresponding to Frisbee's examples provided within his paper, and previously validated test cases from initial examination of this work.

Table 25: Maximum 2D Probability of Collision UnIt Test Cases

Test ID	Description
test01	Alfano test case 1
test02	Alfano test case 2
test03	Alfano test case 3
test04	Alfano test case 4
test05	Alfano test case 5
test06	Alfano test case 6
test07	Alfano test case 7
test08	Alfano test case 8

test09	Alfano test case 9
test10	Alfano test case 10
test11	Alfano test case 11
test12	Manufactured Test Case Corresponding To Frisbee's Example (HBR=5)
test13	Manufactured Test Case Corresponding To Frisbee's Example (HBR=10)
test14	Validation Test Case 1-1 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test15	Validation Test Case 1-3 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test16	Validation Test Case 1-4 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test17	Validation Test Case 1-5 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test18	Validation Test Case 1-6 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test19	Validation Test Case 1-7 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test20	Validation Test Case 1-8 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee

test21	Validation Test Case 1-9 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test22	Validation Test Case 1-10 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test23	Validation Test Case 1-11 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test24	Validation Test Case 1-12 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test25	Validation Test Case 1-13 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test26	Validation Test Case 1-14 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee
test27	Validation Test Case 1-15 from original development (FDSS-II-28-XXXX Single Cov Maximum Pc Validation) validated against independent code base of Joseph Frisbee

2.7 2D Probability of Collision Atmospheric Density Uncertainty Decorrelation

In low earth orbit, predicted covariance matrices include an accommodation for global model error to represent the uncertainty in atmospheric drag modelling on the predicted state of an object. While this aids in a more accurate representation of object state uncertainties at TCA, the fact that there is global uncertainty in these parameters means that the covariance matrices of the primary and secondary objects are correlated and must be decorrelated to accurately calculate Pc.

2.7.1 Atmospheric Density Decorrelation – Mathematical Formulas

Casali et al.[x] previously examined the effect of global model error on probability of collision (Pc) determination and proposed modifications to the Pc formulation to decorrelate the two covariance matrices to better represent the Pc. This process requires that additional information be included on a new generation of CDMs and is currently slated for production and distribution in the near future.

The decorrelation process stems from the combination of the primary and secondary objects in a common frame and removing the cross correlation between these two objects. Typically when calculating 2D Pc the covariance matrices (P) are combined as follows:

$$P_{comb} = P_P + P_S$$

However, to decorrelate the combined covariance, the atmospheric dynamic consider parameter (DCP) density forecast uncertainties (σ_ρ) for each of the two objects, as well as the individual object DCP state sensitivity vectors (G_ρ) must be taken into account. For a more in depth description of the derivation, please see Casali et. al.[x]. Summarization of Casali's Eq. 11 is given below:

$$P_{comb} = P_P + P_S - \sigma_{\rho_P}\sigma_{\rho_S}\left(G_{\rho_S}G_{\rho_P}^T + G_{\rho_P}G_{\rho_S}^T\right)$$

This new combined covariance may then be used in a 2D Pc estimation routine such as those described in Section 2.1 to determine the decorrelated Pc.

2.7.2 Atmospheric Density Decorrelation – Source Code Description

The primary function contained within the SDK used for estimating the decorrelated 2D Probability of Collision of a close approach event is the:

Pc_CovXCorr.m

routine, which estimates the probability of collision using the combined covariance matrix described in the formula above.

As inputs, the routine accepts the following:

Table 26: Decorrelated 2D Probability of Collision Routine Input Parameters

Input Variable	Definition
cdmhead	CDM header structure as returned by function "read_cdm.m" [1X1 struct]
cdmobj	CDM object structures as returned by function "read_cdm.m" [2X1 struct]
HBR	Combined hard body radius or exclusion zone of the two objects (m)
DCPoption	Optional, integer specifying method of retrieving required DCP values. Defaults to option 1 1 – Use DCP Density forecast uncertainty and sensitivity vectors only if specified in the CDM explicitly 2 – Use DCP Density forecast uncertainty and sensitivity vectors if specified in the CDM but use an EDR approximation of these values if unavailable 3 - Use DCP Density forecast uncertainty and sensitivity vectors based on EDR approximations Currently, only option 1 is recommended for use with operational conjunctions. Use of other options is only recommended for development and debugging.
verbose	Optional, binary flag indicating verbose operation, defaults to false

The decorrelated 2D Probability of Collision routine outputs the following:

Table 27: Decorrelated 2D Probability of Collision Routine Output Parameters

Output Variable	Definition
PcXC	2D Probability of Collision calculated using the decorrelated, combined covariance matrix.
CovXC	The 3X3 decorrelated, combined covariance matrix.
DCPvalues	A structure holding the DCP values used in the calculations for both the primary and secondary objects.

Validation cases for this algorithm are contained within the unit test suite for the SDK at:

..\TwoDimensionalPc\UnitTest\CovarianceDeCorrelationCode\Pc_CovXCorr_UnitTest.m

There is currently only a single test case based on an example CDM containing DCP density forecast uncertainty and sensitivity vectors provided by Steve Casali.

Table 28: Decorrelated 2D Probability of Collision Function Unit Test Cases

Test ID	Description
test01	Example CDM with non-zero Pc provided by Steve Casali with relevant DCP density forecast uncertainty and sensitivity vectors.

3.0 Acronyms

BC	Ballistic Coefficient
BFMC	Brute Force Monte Carlo
CARA	Conjunction Assessment Risk Analysis
CDM	Conjunction Data Message
CSpOC	Combined Space Operations Center
DCP	Dynamic Consider Parameter
ECI	Earth Centered Inertial
EDR	Energy Dissipation Rate
GCP	Global Consider Parameter
HBR	Hard Body Radius
LUPI	Length of Update Interval
NPD	Non-Positive Definite
OD	Orbit Determination
ODPO	Orbital Debris Program Office
ODQA	Orbit Determination Quality Assessment
Pc	Probability of Collision
RCS	Radar Cross Section
SDK	Software Development Kit
SEM	Size Estimation Model
SRP	Solar Radiation Pressure
STEVI	Short-Term Encounter Validity Interval
TCA	Time of Close Approach
WRMS	Weighted Root-Mean-Squared

4.0 References

[i] Foster, J. L. and Estes, H. S. *A Parametric Analysis of Orbital Debris Collision Probability and Maneuver Rate for Space Vehicles.* NASA/JSC-25898. Aug 1992.

[ii] Alfano, S. *Satellite Conjunction Monte Carlo Analysis.* AAS 09-233. 2009.

[iii] Elrod, C. Doctoral Dissertation, Department of Statistical Science, Baylor University, 2019.

[iv] Alfano, S. *"A numerical implementation of spherical object collision probability"*. Journal of Astronautical Sciences, 53(1):103–109, 2005.

[v] Hall, D. T., Hejduk, M. D., and Johnson, L. C. "*Remediating Non-Positive Definite State Covariances for Collision Probability Estimation."* AAS/AIAA Astrodynamics Specialist Conference. 2017. Stevenson, WA. AAS 17-567.

[vi] Coppola, V. T., "*Evaluating the Short Encounter Assumption of the Probability of Collision Formula.*" 2012. AAS 12-248.

[vii] Hall, Doyle T., "*Implementation Recommendations and Usage Boundaries for the Two-Dimensional Probability of Collision Calculation*." AAS/AIAA Astrodynamics Specialist Conference, Portland ME. August 2019. AAS 19-632.

[viii] Hejduk, Matthew D., "Satellite Conjunction Assessment Risk Analysis for "Dilution Region" Events: Issues and Operational Approaches" (2019). Space Traffic Management Conference. 28.

[ix] Frisbee, J. H. *An upper Bound on High Speed Satellite Collision Probability when Only One Object has Position Uncertainty Information.* AAS 15-717. 2015

[x] Casali, Stephen J., Hall, Doyle T., Snow, Daniel R., Hejduk, M.D., Johnson, Lauren C., Skrehart, Brent B., and Baars, Luis G. "*Effect of Cross-Correlation of Orbital Error on Probability of Collision*". 2018. AAS 18-272.