Operational Collision Risk Management – Evaluating and Mitigating High-Risk Conjunction Events

Matthew Duncan
SpaceNav

Joshua Wysack
SpaceNav

Operational collision threat characterization is now an essential component of space mission operations. Most spacecraft operators have some semblance of a process to evaluate and mitigate high-risk conjunction events. As the size of the space object catalog increases, satellite operators will be faced with more conjunction events to evaluate. Thus more sophisticated collision threat characterization and collision avoidance strategies must be implemented. This paper presents an overview of SpaceNav's Collision Risk Management software. The software suite enables mission stakeholders to qualify high interest conjunction events. The tools produce various figures and graphs, which aid in analyzing event data. Optimal avoidance maneuver solutions are generated for a user defined set of goals and constraints.

I. INTRODUCTION

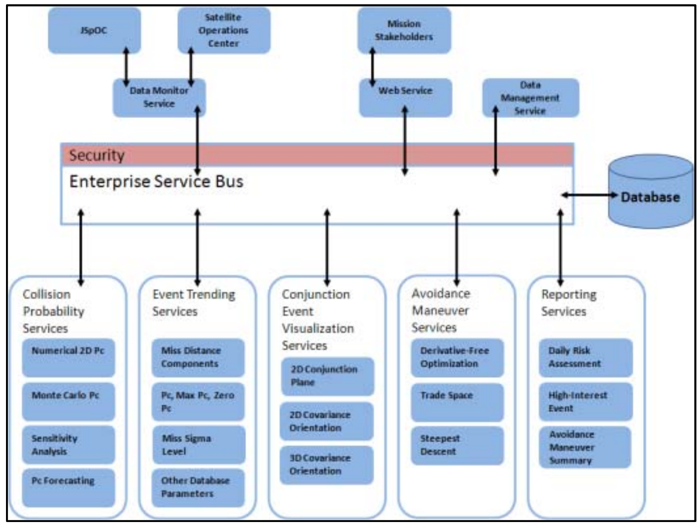
Operational collision threat characterization is now an essential component of space mission operations. The threat characterization process starts with the generation of close approach predictions and ends with an action/no-action decision from mission stakeholders. In particular, the step-by-step process consists of:

1. Reporting all conjunction events that are predicted to violate a specific separation distance threshold over some future time span,
2. Assessing and quantifying the collision threat for each conjunction event that is identified,
3. Developing and executing collision avoidance maneuvers when necessary.

Typical procedures have personnel at the Joint Space Operations Center (JSpOC) performing step 1, and the satellite operator performing steps 2 and 3. As the size of the space object catalog increases, satellite operators will be faced with more conjunction events to evaluate. Low Earth Orbit (LEO) in particular poses unique challenges due to the density of the debris environment, and the effects of atmospheric drag on orbit dynamics. Thus more sophisticated and efficient collision threat characterization and collision avoidance strategies must be implemented.

The following sections present an overview of SpaceNav's Collision Risk Management software. Section II presents the service orientated architecture design and describes the supporting infrastructure. Section III provides details of the different services that comprise the collision threat characterization analysis tool suite. Section IV presents an approach to optimal collision avoidance. Conclusions and a discussion of future work are presented in Section V.

I. SOLUTION ARCHITECTURE



SpaceNav's Collision Risk Management software enables mission stakeholders to analyze and qualify high interest conjunction events.
The software solution is comprised of a set of analysis tools (services), a database, and supporting infrastructure.
Individual functional components are de-coupled so they can be leveraged in a Service Orientated Architecture (SOA) environment.
Automated data processing is performed by the Data Monitoring and Data Management services. The Data Monitor service manages data coming from the JSpOC and data being passed back to the satellite operator. The Data Management services serve as the system orchestrator, and managed the automated data processing and event reporting.
The Collision Risk Management software has been developed in both Matlab and Java. The mathematically intensive analysis algorithms have been developed in Matlab. The supporting data management functionality has been written in Java. Additional details of the major components are provided below:

Enterprise Service Bus
SpaceNav employs a Service Oriented Architecture (SOA) design, utilizing an Enterprise Service Bus (ESB) to manage the services of the collision risk management software. The ESB calls the services through structured messages that pass the required data. This software architecture allows services to be updated without disruption to the entire system, and allows services to be written in different languages; e.g., Java and Matlab. Message adapters are used to ensure that the ESB can communicate with each service. The ESB will have a Message Queue which is a staging or buffering queue that handles messages to and from services. It can resubmit messages that are initially rejected.

Analysis Services
The analysis functions are represented as de-coupled, individual processes or services within the overall architecture. Key analysis services include: collision probability analysis, collision avoidance planning, plot generation, and reporting. Services are called through messages sent by the ESB. A Data Monitor Service will continually watch for new data to be posted to the server. Once new data is made available, the Data Monitor Service will process the data and generate reports as necessary. Automated daily conjunction summary reports will be generated based on a subset of this data. As part of the automation process, these reports will be generated and sent to the appropriate recipients upon completion.

Database
A MySQL database is utilized to store all conjunction event data. This data includes conjunction geometry parameters, state & state uncertainty, and a set of derived parameters that are calculated by the services. The MySQL database provides a secure and efficient means to store and access data. The database will communicate with the ESB through a Java Database Connectivity (JDBC) Application Program Interface (API).

Web Services
Web services are utilized to deliver data products to the mission stakeholders. Products include daily conjunction summary reports and high-interest event reports.

The following sections provide functional details and description of the key analysis services within the Collision Risk Management software. In particular, a description of the collision threat characterization and optimal collision avoidance tools are presented.

II. COLLISION THREAT CHARACTERIZATION

A. Operational Overview

Daily close approach predictions are produced by Air Force personnel at the JSpOC at Vandenberg AFB. If two objects are predicted to come within some separation threshold, JSpOC personnel will issue a warning report and notify the appropriate satellite operator. Additionally, the JSpOC will provide various supplementary data products to the operator so that the collision threat can be established. Mission analysts must make sense of the JSpOC data by producing trends, comparing data for statistical consistency, and eventually quantifying the risk of collision [1].

The primary metric used to assess and quantify the collision risk is the probability of collision. If the collision probability is high then the current risk is high. Low probability cases fall into two different categories: They are neither a current nor a future risk, or they will become a high risk event at some future time. In both the high probability and low probability cases, characterizing the current and future collision risk is challenging.

For example, assume a given event has a collision probability value of 1 chance in 5000. This would be considered a high risk event. However, this event may not be high risk in the future. One or both objects may have a large covariance; the prediction time may be several days in the future, etc. So even though the event is characterized as 'high risk', there is a chance that the risk will subside. Conversely, a low probability event may evolve into a high risk event as the time to closest approach is shortened. The change from a low risk to a high risk event could be caused by additional tracking data being collected on one or both objects.

The fact that the risk can (and most likely will) change is at the heart of the operational collision threat characterization problem. The Collision Risk Management software has been designed to aid in the analysis of the risk so that the current and future collision threat can estimated. The Collision Risk Management analysis utilities span 5 different services:

B. Analysis Services

- *Collision Probability Service*: The Collision Probability Service computes various collision risk metrics. The 'traditional' high speed 2-D collision probability is computed [2]. Moreover, a Monte Carlo simulation is available for more complex orbit geometry such as slow-speed or formation flying encounters. Additional metrics such as the maximum probability and a zero miss probability are also generated.

- *Event Trending Service*: The Event Trending Service produces time history trends for a given conjunction event. Relevant parameters such as the collision probability, combined position uncertainty and sigma level ratios are plotted over time.

- *Visualization Service*: The Visualization Service generates a range of figures showing the two objects' relative orientation and conjunction geometry at the point of closest approach. For example, the combined covariance projected into the conjunction plane shows the covariance orientation relative to the safety keep-out region.

- *Event Reporting Service*: The Event Reporting Service produces .pdf and PowerPoint files that contain both text and graphical output produced from the other analysis services. A mixture of different text and graphics data can be combined to give a customized report.

- *Collision Avoidance Maneuver Service*: The Collision Avoidance Maneuver Service generates risk reduction maneuvers, targeting one or more conjunction events. The Avoidance Maneuver Service allows the user to choose three different methods when planning a maneuver. Additional information on one of the avoidance algorithms is presented in Section III.

III. OPTIMAL COLLISION AVOIDANCE

Once a high-risk conjunction event is identified, various avoidance scenarios are generated. An acceptable maneuver plan must produce an orbit change that reduces the collision risk, while still meeting various operational constraints. Common operational constraints include restrictions on: maneuver magnitude, maneuver direction, and the time of day when the maneuver can be executed. Often times, avoidance planning methods do not specifically target or even directly consider more than a single conjunction event at a time.

Typical avoidance maneuver planning procedures create a maneuver that reduces the collision risk for the highest probability event. The aggregate residual risk that spans all new conjunction events is not accounted for directly, since trajectory information for the space object catalog is not readily available. Consequently, a given maneuver plan is validated by running the proposed trajectory through the entire JSpOC conjunction assessment (CA) process, so that all new conjunction events are identified. Then, the collision threat characterization process starts from the beginning again. The maneuver must be chosen so that no new high-risk events are created.

If it turns out that a new high-risk event is introduced, subsequent adjustments to the maneuver profile must then be made to account for both high-risk events. For each event, the 'standard' 2-D collision probability, is computed [2] .Additionally, the avoidance planning process must account for large uncertainty 'monitor' events. The iterative 'back and forth' with the JSpOC can be cumbersome and inefficient when trying to evaluate multiple maneuver plans. To ensure that the total risk of the event maneuver plan is quantified, a metric is employed to account for all conjunction events. The measure of total risk as defined by multiple events is determined by taking into account the 2-D collision probability from each individual event. The compliment of P_c is the probability of there not being a collision. To find the aggregate probability of collision, the probability of no single collision amongst all events if found, and the compliment is taken. Aggregate collision probability can then be defined as:

$$P_c^A = 1 - \prod_{i=1}^{M} \left(1 - P_{c_i}\right),$$

where M is the number of conjunction events. To find an optimal avoidance maneuver that mitigates the risk posed by multiple conjunction events a derivative-free optimization (DFO) technique is utilized. The following section will introduce the DFO method. First, the details the mathematical details of DFO will be presented, followed by the application of DFO to the avoidance maneuver planning problem.

A. The DFO Algorithm

The DFO method is an unconstrained, multi-variable optimization technique. The algorithm iteratively forms a quadratic model that produces a local approximation of the cost function. Constraints are simply added to the cost function to apply additional cost or 'penalty' as required. The DFO algorithm operates on a bounded region called the trust region. Figure 1 below shows the DFO algorithm. Additional details are provided in the following sections.

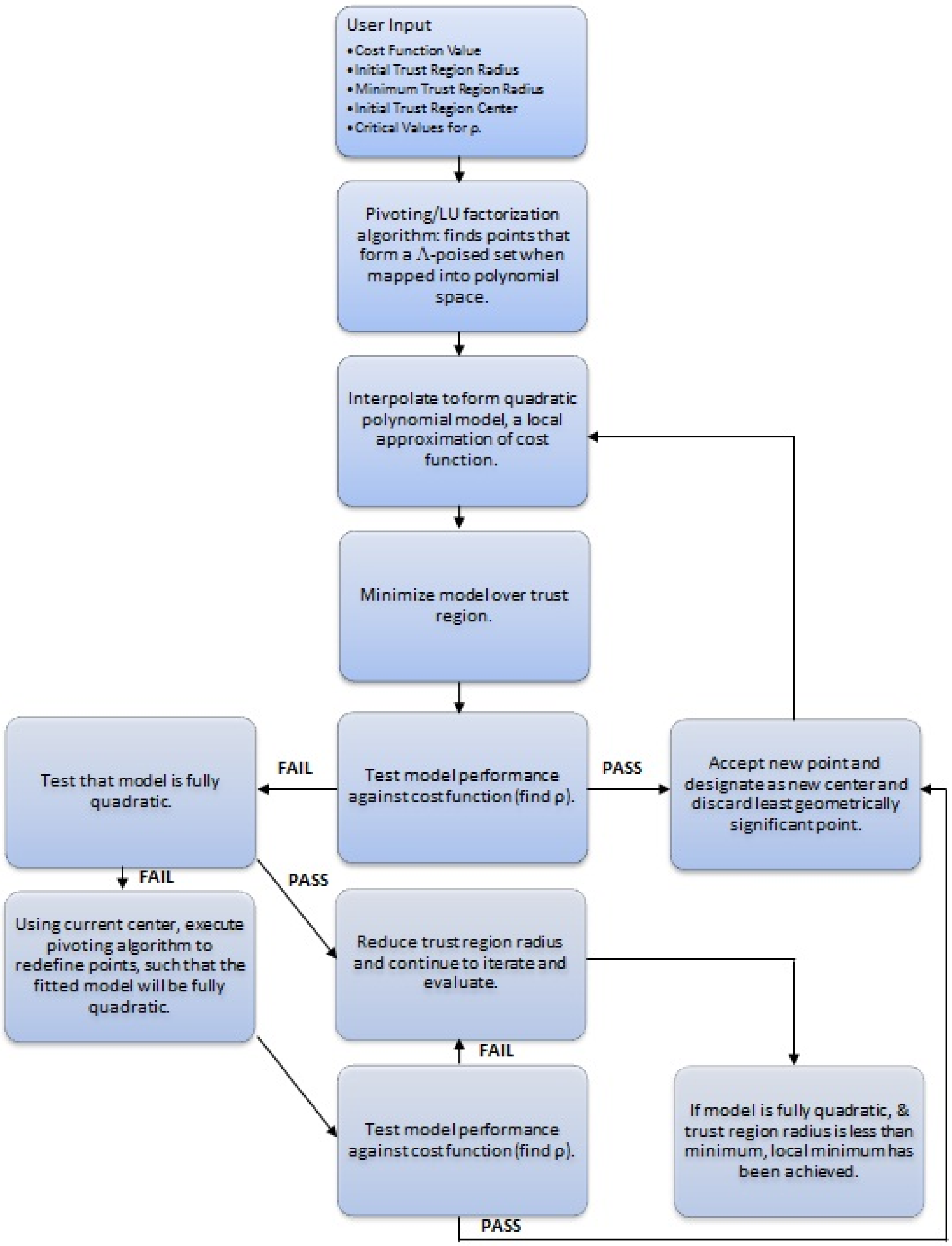


Figure 1: DFO Algorithm - Summary

B. Trust Region, Λ-Poised and Fully Quadratic

The terms trust region, Λ-poised, and fully quadratic, as referred to in Figure 1, will now be defined. The DFO method minimizes the multi-variable cost function, *f*, over a bounded region, called the trust region. The trust region is a ball with radius Δ centered at point *x* and denoted by B(x; Δ). To form a model that approximates the cost function locally, the algorithm is restricted to evaluating points within the trust region. When the model is minimized, it is minimized over the trust region. As the DFO algorithm converges to a local minimum, Δ will decrease to a user specified lower bound of Δ_{min}.

The number of points required to form a unique quadratic function is defined to be P = (n+1)(n+2)/2. An initial center point is chosen and the remaining points will be determined using a pivoting algorithm such that, when these points are mapped into polynomial space, they form a *P*x*P* non-singular matrix. A set of points that, when mapped into polynomial space, forms a nonsingular matrix will be referred to as Λ-*poised*, where poised is the term used to describe how well the set spans the region of interest[3].

Additionally, a requirement of the DFO algorithm is that the cost function be smooth. Specifically, the DFO algorithm requires that the cost function be twice continuously differentiable, with Lipshitz continuous Hessian over some open domain, appropriate for the cost function being evaluated. Lastly, the term fully quadratic implies that the preceding requirements pertaining to smoothness have been met and that:

the error between the Hessian of the model and the Hessian of the cost function satisfies

$$\|\nabla^2 f(y) - \nabla^2 m(y)\| \leq \kappa_{eh}\Delta \quad \text{for all } y \in B(x;\Delta),$$

the error between the gradient of the model and the gradient of the cost function satisfies

$$\|\nabla f(y) - \nabla m(y)\| \leq \kappa_{eg}\Delta^2 \quad \text{for all } y \in B(x;\Delta),$$

the error between the model and the cost function satisfies

$$\|f(y) - m(y)\| \leq \kappa_{ef}\Delta^3 \quad \text{for all } y \in B(x;\Delta),$$

where,$\kappa_{eh}, \kappa_{eg}, \kappa_{ef}$ are positive constants. Moreover, for any *x* within the maximum trust region neighborhood and $\Delta \in (0, \Delta_{max}]$, there exists some model *m*(*y*), with Lipschitz continuous Hessian and corresponding Lipschitz constant bounded by some positive constant. A Lagrange interpolation model built on a Λ-poised set will necessarily yield a fully quadratic model [3].

C. The DFO Algorithm Process

The algorithm starts with an initial point x_0 and an initial trust region. For each DFO iteration, a new point is found by minimizing the quadratic polynomial over the trust region. Once a new point is found, it must be tested with the cost function to determine if descent has occurred. If the model accurately predicts a descent in the cost function the point is accepted as the new center and the process repeats. The algorithm is presented in a series of steps below.

Step 0. Initialization: Choose an initial point, initial trust region, minimum (convergence) trust region. Form initial model.

Step 1. Criticality Step: Test that model is fully quadratic. If model is fully quadratic, keep current model and continue; if not, form new model.

Step 2. Step Calculation: Minimize the model over the trust region to form trial point. The distance between the current point and the minimum point is the current step, s_k.

Step 3. Acceptance of Trial Point: Calculate the performance metric,

$$\rho = \frac{f(x_k) - f(x_k + s_k)}{m(x_k) - m(x_k + s_k)}$$

If $\rho \geq \eta_1$ or $\rho \geq \eta_0$ and the model is fully quadratic, then accept new point. Designate point as new center and discard least geometrically significant point. (This step will be discussed further below.)

Step 4. Model Improvement: If $\rho < \eta_1$and model is not fully quadratic, improve model.

Step 5. Trust Region Radius Update: If $\rho \geq \eta_1$, increase the trust region radius by a factor of 2.
If $\rho < \eta_1$and model is fully quadratic, decrease the trust region radius by a factor of ½. Otherwise, keep the same trust region radius for next iteration. Return to Step 1 [4].

D. Evaluating the Performance of the Model

The ratio of the observed decrease in the cost function to the decrease expected by the model, denoted by ρ, is the metric used to measure how well the model is approximating the actual behavior of the cost function.

If ρ is large, $(\rho \geq \eta_1 = 0.5)$, the model is performing well and the point will be accepted and used as the center for the next iteration. Because the model is generated using interpolation, which does not allow for over-determined systems, a point must be discarded, so that the new point can be incorporated into the set. The DFO algorithm will discard the least geometrically significant point to accomplish this. The trust region will be expanded; this implies that while the model is performing well, it may increase the step size, s_k, at each iteration.

If ρ is small $(\rho < \eta_1)$, the model did not accurately predict the behavior of the cost function. If this occurs, the DFO algorithm will perform a series of checks, the first being that the model is fully quadratic. If the model is not fully quadratic, the algorithm will discard points and replace them using the pivot method, a type of LU factorization, about the current center. Once the set is Λ-poised, the algorithm forms a new model which is now certified fully quadratic.

After the model has been certified fully quadratic, if ρ is still small, the algorithm will begin to reduce Δ. It will continue to minimize the model over the reduced trust region, and evaluate the point with the cost function, testing for increases in ρ. If the model is fully quadratic, and Δ becomes less than Δ_{min}, the model has reached a local minimum, and thus found the solution [4].

IV. DFO APPLIED TO AVOIDANCE MANEUVER PLANNING

This section presents the formulation of the optimal collision avoidance problem, with constraints.
The DFO algorithm varies both maneuver time and magnitude in order to produce a maneuver plan that reduces the aggregate collision probability to a specified value. Modification of the cost function to handle this constraint and others will be presented. When the collision probability is taken as an aggregate over multiple conjunction events, no analytic derivative information is available. The unconstrained cost parameter for this problem is the maneuver magnitude. Since that value alone does not specify the goal of mitigating a high-risk conjunction event, constraints must be imposed. The primary constraint is the reduction of the P_c^A below the designated threshold. A P_c^A value below the threshold implies that the P_c value for each conjunction event is also below the desired threshold. In order to frame the problem as an unconstrained optimization problem, as is required by the DFO algorithm, constraints must be included as part of the cost function. This is done by forming penalty functions that impose additional cost when the constraint is violated. The optimization problem can be stated as the minimization of the cost function,

$$f = |dv| + \sum_{i=1}^{M} \alpha_i (g_i)_+ \ \acute{\alpha} \sum_{i=1}^{N} \alpha_i (\acute{g}_i)_+$$

Where the notation $(\acute{g}_i)_+$ represents

$$(\acute{g}_i)_+ = \begin{cases} 1 & \text{for } \acute{g}_i > 0 \\ 0 & \text{othewise} \end{cases}$$

Here, g_i represents the i^{th} penalty function, and α_i represents the associated i^{th} weight used to penalize solutions that violate the corresponding constraint. These are 'distance-based' penalty functions that apply a penalty proportional to the distance of the current point from the region of feasible solutions. The $\acute{g}_i$ represent 'static' penalty functions that impose the same penalty whenever they are violated [5]. Frequently, they can be calculated directly from the current point. The associated weight, $\acute{\alpha}_i$, is imposed on the cost function for violation of these constraints. The constants M and N represent the number of distance-based and static penalty functions, respectively. The P_c^A threshold is enforced through a distance-based penalty function expressed as,

$$g_1 = log_{10}(P_c^A) - log_{10}(P_T),$$

for P_c^A and threshold P_T . Maneuver magnitude and maneuver time constraints are imposed with static penalty functions; they are implemented as,

$$\acute{g}_{1_1}' = dv_{min} - dv \ \ \& \ \acute{g}_2 = dv - dv_{max}$$

$$\acute{g}_3 = t_{dvMin} - t_{dv} \ \ \& \ \acute{g}_1 = t_{dv} - t_{dvMax}$$

where the minimum and maximum constraints represent bounds on the maneuver. The process is initiated when the DFO algorithm provides the current point (maneuver magnitude and time) to orbital dynamics software that will apply the maneuver to alter the trajectory of the Primary satellite. At this point the same software is used to calculate the value of parameters needed for evaluation of mission unique constraints. An example parameter is the satellite-earth-sun angle at burn time. These values will be used in the evaluation of the associated penalty functions. The orbital dynamics software will be used to create ephemeris and covariance files for the new trajectory. Next, points of close approach will be found by comparing the Primary's new ephemeris against ephemeris for a Secondary object (this data is made available by the JSpOC). The P_c calculation is then performed for each close approach that violates a specified miss distance. These steps are performed for each Secondary object. Once all of the individual P_c values have been calculated, the P_c^A calculation can be performed. Finally, the cost function is evaluated using the P_c^A value and any constraint violation parameters that are used in the penalty functions. The value of the cost function is returned to the DFO algorithm.

V. CONCLUSIONS & FUTURE WORK

As the size of the space object catalog increases, satellite operators are faced with more threats to evaluate. Thus the risk characterization and strategies and risk reduction solutions become even more complex. SpaceNav's Collision Risk Management software provides the satellite operator with several tools designed to qualify high interest conjunction events. The tools produce various figures and graphs, which aid in analyzing event data. If a high risk event is detected, the software will generate an optimal collision avoidance maneuver. Future work will consist of continued research and testing of the probability forecasting method.

VI. REFERENCES

[1] A. Bleich, M. Duncan, and J. Wysack, "The Collision Risk Assessment & Risk Mitigation Process For the NPP & NPOESS Missions," AAS 09-375.

[2] J. Foster, "The Analytical Basis for Debris Avoidance Operations for the International Space Station and Space Shuttle," Orbital Debris Quarterly News, Vol. 6, No. 2, 2001, p. 1.

[3] A. Conn, K. Scheinberg, and L. Vicente, Introduction to Derivative Free Optimization. Society for Industrial and Applied Mathematics and the Mathematical Programming Society, 2009.

[4] S. Billups, J. Larson, and P. Graf, "Derivative-Free Optimization of Expensive Functions with Computational Error Using Weighted Regression," .

[5] T. Baeck, D. Fogel, and Z. Michalewicz, Handbook of Evolutionary Computation. Taylor and Francis Group, 1997.

[6] G. Peterson, "Maneuver Selection for Probability Reduction of Near-Circular Orbit Conjunctions," AAS 2002-4630.