(Preprint) AAS 21-717

ANALYSIS OF POTENTIAL COLLISION MANEUVER GUIDELINES FOR FUTURE SPACE TRAFFIC MANAGEMENT

Richard J. Macke,[*] Brian C. Gunter,[†] Mariel Borowitz[‡], and Megan Birch[§]

The growing requirement for a space traffic management system (STM) has prompted exploration of potential STM rules and regulations. To evaluate these proposals, a model must be developed that captures both the current behavior or resident space objects (RSOs) and their behavior under future STM guidelines. This paper discusses the development of such a model and simulation environment to examine proposed STM regimes by producing the impact on conjunctions between RSOs and the cost to satellite operators for each proposal. These are the costs and benefits that will be weighed when determining the effectiveness of each potential STM regime.

INTRODUCTION

Since the first artificial satellite was launched in 1957, the quantity of Resident Space Objects (RSOs) in orbit has continued to increase. These objects include operating satellites, rocket-bodies used to insert objects into orbit, inoperative satellites, and debris from launches, satellite break-up, and collisions between two on-orbit objects. Excluding operating satellites, these objects are often considered orbital debris. Due to the large magnitude of the relative velocities found in orbit, any collision involving an operating satellite would result in the satellite's destruction. Furthermore, any collision in orbit would generate more debris that could threaten operational satellites.

In 2009, the inactive Russian satellite Cosmos 2251 collided with the active U.S. satellite Iridium 33. This collision produced the second largest amount of space debris ever recorded, at least 2,200 fragments. Additionally, a test of an antisatellite weapon by China produced over 3,400 fragments. These two events alone increased the number of detectable objects in orbit by approximately 65 percent.[1] Adding to this are the tens of thousands of new satellites predicated to be in orbit in the next two decades as numerous planned constellations are implemented, to include several global broadband megaconstellations.

Because of the increasing number of RSOs, debris or not, work is needed to develop a space traffic management (STM) system that comprises of policies and procedures to prevent future on-orbit collisions. STM can be defined as "the planning, coordination, and on-orbit synchronization of activities to enhance the safety, stability, and sustainability of operations in the space

[*] Graduate Research Assistant, Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, 270 Ferst Drive, Atlanta NW, Georgia 30332-0150
[†] Assistant Professor, Daniel Guggenheim School of Aerospace Engineering. Georgia Institute of Technology, 270 Ferst Drive, Atlanta NW, Georgia 30332-0150, and AIAA Associate Fellow.
[‡] Associate Professor, Sam Nunn School of International Affairs, Georgia Institute of Technology, 270 Ferst Drive, Atlanta NW, Georgia 30332-0150, AIAA Associate Fellow
[§] Research Scientist, Electro-optical Systems Laboratory, Georgia Tech Research Institute, 250 14th Street, Atlanta NW, Georgia 30318-5394

environment." Such a system will be an important part of keeping low Earth orbit (LEO) sustainable for the future as activity in orbit continue to grow.[2]

In an effort to test the effects of different STM regimes, we developed a flexible simulation environment of on-orbit objects to estimate their possible collisions (conjunctions). This simulation environment provides for the analysis of extensive catalogues of on-orbit objects for long periods of time. It also allows for the detection of conjunctions between objects based on user-specified criteria such as the minimum range between two objects passing on another, or the probability of collision of the two objects. While the conjunction analysis generates results and metrics with which certain STM regimes can be assessed, the implementation of prospective STM rules and regulations in the simulation is done by virtually maneuvering the satellites in accordance with the provided STM procedures. This maneuvering schema attempts to approximate the behavior of satellites on-orbit while remaining computationally efficient. Various metrics are gathered throughout each simulation to assess the overall effectiveness of each set of STM procedures, to be detailed later.

ORBITAL PROPAGATION

The basis for the simulation is the propagation of objects in orbit. Due to the number of possible RSOs being simulated and the time windows of simulation, the Simplified General Pertubations-4 (SGP4) model was used. SGP4 accounts for perturbations caused by Earth's non-spherical shape, drag, radiation pressure, and the gravitation of third bodies[3]. The benefits of the SGP4 model include its relative computational efficiency and its analytic nature. By being analytical in nature, SGP4 calculates the state, position and velocity vectors, of an object in similar computational time for epochs both relatively close to and relatively far from the initial epoch, e.g., the calculation of the state of an object one minute from the provided initial state values takes similar time to accomplish as the calculation of the state of an object 5 days from the provided initial state values. This reduces the time to propagate objects over large time windows compared to numerical propagators that calculate the future state of an object at a certain epoch using the state of the object one time step before the propagation epoch. However, the simplified nature of SGP4 means that its error increases by approximately 1 to 3 kilometers per day.[4]

SGP4 was created by the North American Aerospace Defense Command (NORAD) for use with their two-line elements (TLE) data format. Three-line elements are TLEs with an additional line at the beginning that is the name of the object given by the United States Space Force's 18th Space Control Squadron (18SPCS). TLEs themselves are two fixed-length lines of text that give information about an orbital object's state at a certain epoch. TLEs are also intended to be used as inputs for the SGP4 propagator. Additionally, a catalogue of all data tracked by NORAD, except for satellites that NORAD excludes for national security purposes, is available online* in TLE form from the 18SPCS. [3] Therefore, the simulation expects an input file consisting of a list all the objects to be included in TLE form or three-line element form.

Because the simulation is intended to be used to analyze the effects of STM regimes on large catalogues of data, and not to be a precise predictor of actual orbital conjunctions, SGP4 is assumed to sufficiently model the behavior of orbital objects, i.e., the results of SGP4 are assumed to be completely accurate in the "universe" of the simulation.

* https://www.space-track.org

CONJUNCTION ANALYSIS

To evaluate the effectiveness of an STM regime, one of the most important metrics is the reduction in the number of conjunctions found after the regime is implemented compared to other STM systems, or the complete lack of one. Therefore, the simulation includes a procedure to find conjunctions between all objects in the simulation.

Finding conjunctions requires comparing each RSO to every other RSO at each time step to determine whether the pair of RSOs is within a defined maximum range of closest approach, also known as R_{cr}, or if they passed within this range between the previous time step and the current one. However, calculating the range of closest approach within a certain time step for each pair of RSOs will expend large amounts of computing time and resources. So, a method of filters is implemented to remove RSO pairs that could not possibly pass within the given maximum range of closest approach before the next time step. These filters are increasing in complexity to filter out as many RSO pairs as possible using the smallest amount of computation time.

The first filter, a common one used in many conjunction analysis filtering schemes, is a perigee-apogee filter. This filter looks at the perigee and apogee of each RSO in a pair. If the perigee of one RSO is larger than the apogee of the other by greater than the R_{cr}, there is no way for the two RSOs in question to conjunct. Equation (1) is used to determine whether a pair of RSOs pass or fail this filter. Because this filter only implements four operations, it is relatively quick, and thus is used as the first filter.

$$\max\left(perigee_1, perigee_2\right) - \min\left(apogee_1, apogee_2\right) > R_{cr} \tag{1}$$

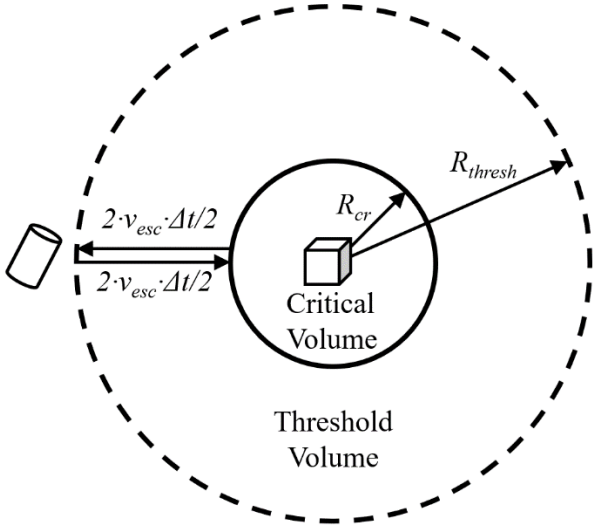

Figure 1. Relationship between threshold volume and R_{thresh}, the critical volume and R_{cr}, and the maximum possible relative velocity (v_{esc}) (adapted from [5]).

For further filtering, a variation of the "Smart Sieve" method is used.[5] This method uses an implementation of three filters. The first filter is based on the designation of a threshold volume. Designating one RSO the secondary and one the primary and centering a frame of reference on the primary object, the threshold volume is a sphere around the primary such that a secondary RSO cannot enter the critical volume, a sphere around the primary with a radius of R_{cr}, and then leave the threshold volume within the same time step. See Figure 1 for a graphical representation.

Equation (2) shows the calculation for the threshold radius, R_{thresh}, where v_{esc} is Earth's escape velocity and Δt is the propagation time step. The r^2 filter is the filter that removes RSO pairs that are farther than R_{thresh} away from each other. This is done by removing RSO pairs such that for relative position, $\boldsymbol{r}_0$, $r_{0,x}^2 + r_{0,y}^2 + r_{0,z}^2 > R_{thresh}^2$.

$$R_{thresh} = R_{cr} + v_{esc} * \Delta t \tag{2}$$

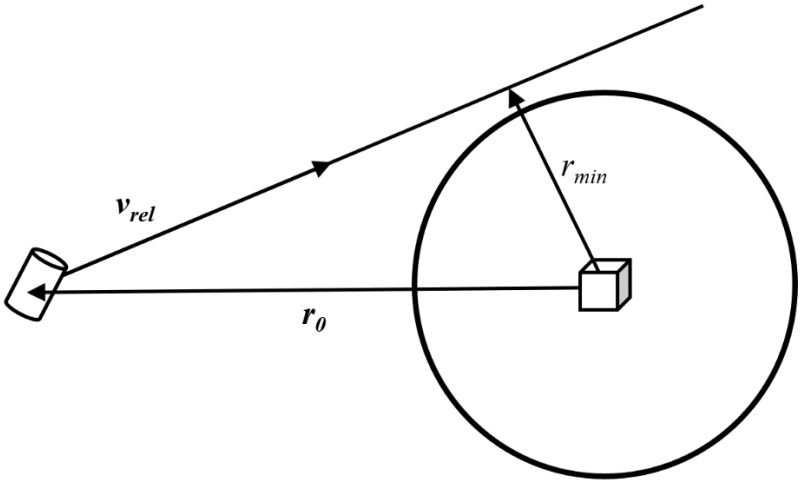

Figure 2. Definition of r_{min} from relative velocity vector (v_{rel}) and relative position vector (r_0), (adapted from [5]).

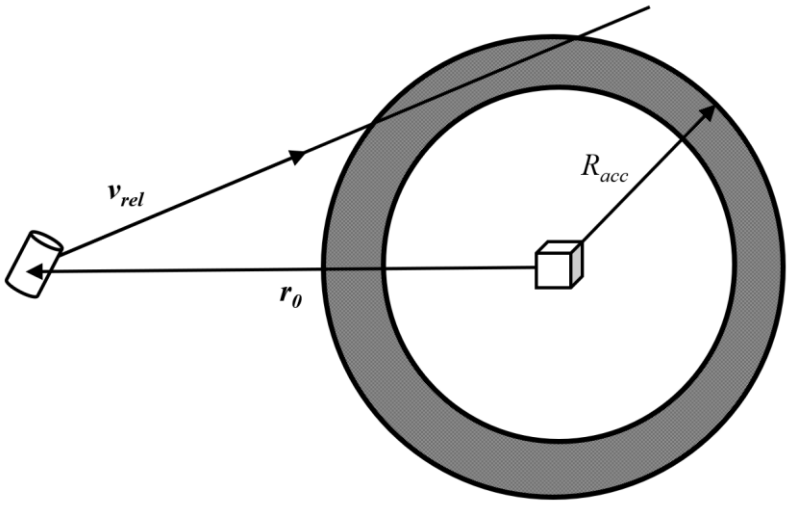

Figure 3. New critical volume, its radius (R_{acc}), and its relationship to the relative velocity vector (v_{rel}) and the relative position vector (r_0) (adapted from [5]).

Next, the range of closest approach can be estimated, excluding the orbital bending due to gravity. Equation (3) shows how to calculate this range of closest approach estimate, r_{min}. The next filter takes advantage of the fact that the relative acceleration between two RSOs can never be more

than twice the gravitational acceleration at sea level. Combining the maximum difference between a linear path and the true relative motion of the secondary with respect to the primary, with the critical volume, a new critical volume is created. The radius of this new critical volume, R_{acc}, is found using Equation (4). The second sieve combines these two, and discards RSO pairs that have an $r_{min} > R_{acc}$. Figures 2 and 3 show the geometry of this filter.

$$r_{min} = \sqrt{r_0^2 - \left(\boldsymbol{r}_0 \cdot \frac{\boldsymbol{v}_{rel}}{v_{rel}}\right)^2} \tag{3}$$

$$R_{acc} = R_{cr} + g_0 \Delta t^2 \tag{4}$$

Finally, the last filter calculates another threshold radius. However, this filter uses the actual relative velocity instead of twice Earth's escape velocity. This filter requires more computation, which is why the less accurate, but simpler threshold filter is used first. Equation (5) shows how to find the new, finer threshold volume radius, $R_{thresh,fine}$. This filter uses the same test as the first one, replacing R_{thresh}^2 with $R_{thresh,fine}^2$: $r_{0,x}^2 + r_{0,y}^2 + r_{0,z}^2 > R_{thresh}^2$.

$$R_{thresh,fine} = R_{acc} + \frac{1}{2}\left(\boldsymbol{v}_{rel} \cdot \frac{\boldsymbol{r}_0}{r_0}\right)\Delta t \tag{5}$$

The last step, after discarding as many RSO pairs as possible, is to calculate the actual range of closest approach. For this step, an iterative process uses a Taylor approximation that takes an initial guess of the time of closest approach (TCA). The next TCA estimation is formed using the relative position and relative velocity vectors from the previous step. For each iteration, SGP4 is called at the new TCA estimate. After the TCA value converges, the final TCA is input into the SGP4 propagator to find the two states at TCA, then the actual range of closest approach. The iteration scheme is shown in Equation (6). The iteration is run until the time converges within a threshold value. This value is defined as the time an object would take to travel 50 meters if it were travelling at the relative velocity between the two RSOs. The 50-meter error margin was chosen as a value that allowed the iteration schema to converge rapidly while still providing a high level of accuracy when determine the range of closest approach.

$$t_{i+1} = t_i - \frac{\boldsymbol{r}_{0,i} \cdot \dot{\boldsymbol{r}}_{0,i}}{\dot{\boldsymbol{r}}_{0,i} \cdot \dot{\boldsymbol{r}}_{0,i}} \tag{6}$$

Although the conjunction determination function is run at every propagation time step, it does not necessarily check every possible RSO pair at every step. If the distance between two RSOs is much greater than the R_{cr}, the secondary object will not come within R_{thresh} for a number of steps. So, comparison between the two RSOs can be skipped for multiple time steps. Equation (7) gives a conservative number of steps to skip that will not result in skipping a future conjunction.

$$N_{skip} = \text{int}\left(\frac{r_0 - R_{thresh}}{2v_{esc}\Delta t}\right) \tag{7}$$

This number is then stored for each RSO pair. If it is greater than 1, the RSO pair is not compared.

IMPLEMENTING MANEUVERS

The next step after discovering a conjunction is to maneuver the RSOs, if possible, to eliminate the conjunction. For a certain conjunction, a corresponding maneuver is implemented that alters

the orbit, or orbits, of the RSO(s) such that the conjunction no longer occurs. Currently, the maneuver used is an orbital phasing maneuver. This maneuver adjusts the time of perigee while leaving the RSO in the same orbital track. This changes the time at which one of, or both of, the RSOs will reach the conjunction point. Equations (8) through (13) show the calculations to determine the change in velocity required to enter an orbit that will move the RSO by an angle of ϕ in its orbit. The direction of the change in velocity will align with the RSO's current velocity vector. After an orbit, a change in velocity (δv) of the same magnitude in the opposite direction is imparted. After the conjunction has passed, the process is performed again with a negative value of the original ϕ. This process is updated slightly to increase the period between the first and second burns to be as many orbits as possible before the conjunction instead of using only a single orbit to coast. This reduces the δv required for both burns. Figure 4 shows a phasing maneuver graphically.

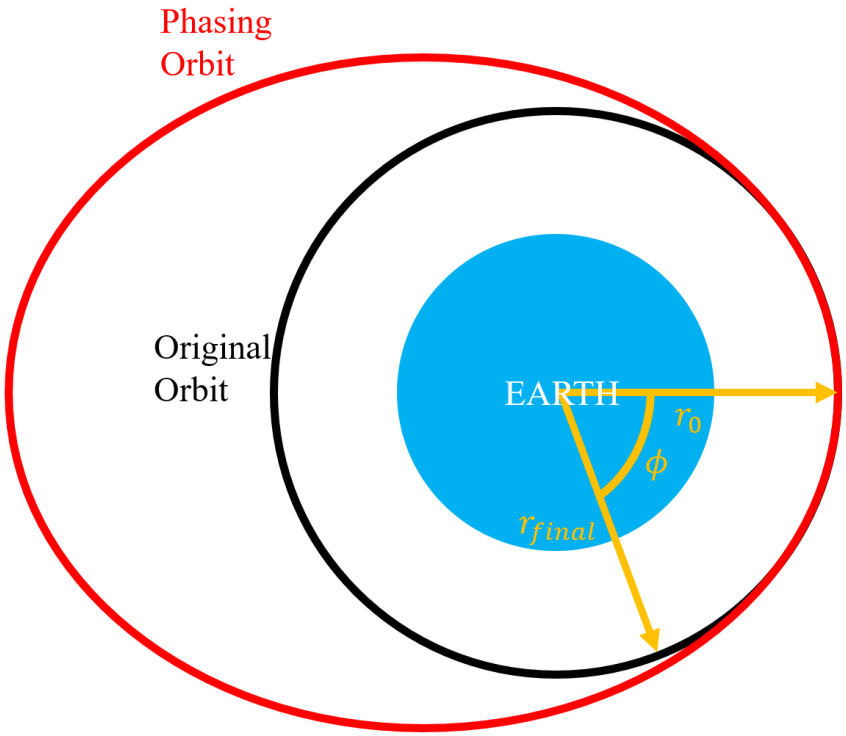

Figure 4. Orbit phasing maneuver. At r_0, spacecraft increases velocity to raise apogee. In phasing orbit, spacecraft takes longer to complete a single orbit, so once it reaches perigee after an orbit, it decreases velocity to lower apogee back to original altitude before maneuver. The size of the phasing orbit determines how far behind the spacecraft lags, and thus ϕ.

$$T_1 = 2\pi\sqrt{\frac{a_1^3}{\mu}} \tag{8}$$

$$\delta E = 2\tan^{-1}\left(\sqrt{\frac{1-e}{1+e}}\tan\left(\frac{\phi}{2}\right)\right) \tag{9}$$

$$tphase_1 = \frac{T_1}{2\pi} * (\delta E - e * \sin(\delta E)) \tag{10}$$

$$T_2 = T_1 + tphase_1 \tag{11}$$

$$a_2 = \left[\sqrt{\mu}\left(\frac{T_2}{2\pi}\right)\right]^{\frac{2}{3}} \tag{12}$$

$$\delta v = \sqrt{\mu\left(\frac{2}{r_0} - \frac{1}{a_2}\right)} - v_0 \tag{13}$$

After determining the δv required, the satellite should be reinitialized with a state equivalent to the old velocity vector plus the δv and the same position as before. However, SGP4 does not take position and velocity as state values. Instead, it takes orbital element values as an input. The most complicating factor, though, is that it does not take the normal orbital elements that can be calculated from position and velocity, sometimes called oscillating orbital elements, but instead takes mean orbital elements. So, a method to convert oscillating orbital elements to mean orbital elements must be used.

This simulation uses a direct iteration approach to find the mean orbital elements that correspond to given oscillating orbital elements.[6] The process begins by using the oscillating orbital elements as the initial guess for the mean elements. Then, run SGP4 with those inputs to find new state vectors. Take the difference between the newly calculated state vectors and the original state vectors and add that to the previous iteration's guess. That will be the next iteration's guess. Convert those state vectors into oscillating orbital elements and repeat. Equations (14) through (17) describe this process. $\boldsymbol{x}$ represents the state vector, position and velocity combined, and the $OrbElements$ are the oscillating orbital elements calculated from the state vector.

$$\boldsymbol{x}_{sgp,i} = SGP4(OrbElements_i) \tag{14}$$

$$\delta\boldsymbol{x}_i = \boldsymbol{x}_0 - \boldsymbol{x}_{sgp} \tag{15}$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \delta\boldsymbol{x}_i \tag{16}$$

$$OrbElements_{i+1} = State2OrbElements(\boldsymbol{x}_{i+1}) \tag{17}$$

This iteration is run until each component of the position and velocity vector produced by the estimated mean orbital elements is within 10^{-7} km or km/s of the desired value.

After determining the necessary mean orbital elements, the RSO that is intended to maneuver is reinitialized in its new state using the calculated elements. Every time a maneuver is implemented the iteration must be run to determine the values that will put the RSO in its desired state. Fortunately, this iteration scheme converges quite quickly. So, it is not a bottleneck in the simulation that increases runtime significantly. Additionally, after TCA has passed, the simulation has the option to maneuver the RSO back to its original position.

MODELING REALISTIC MANEUVER BEHAVIOR

Now that the infrastructure to implement maneuvers has been developed, the next step is to determine the magnitude of the maneuver to implement. Because only an orbital phasing type of maneuver is built, the magnitude is measured by the angle ϕ by which to move the RSO. The first step is to choose a starting ϕ that ensures the conjunction in question is avoided. The ϕ value is chosen to be the angle that is subtended by an arc with a length equal to R_{cr} at a radius equal to the RSO in question's orbital radius at TCA. This angular value is multiplied by a conservative safety factor of 2.5. Equation (18) shows the equation used to calculate this ϕ value.

$$\phi = 2.5 * \frac{R_{cr}}{r_{TCA}} \tag{18}$$

While this choice of ϕ will avoid the original conjunction, it may create future conjunctions. So, a method for adjusting the ϕ value is needed to choose one that will not create future collisions with other RSOs. We named this process the "sufficing" process as it is unlikely an automated process that runs sufficiently quickly to allow for large simulations will determine the most optimal maneuver for every conjunction. However, this process suffices in determining a maneuver that will avoid future collisions with other satellites.

The sufficing process in the simulation starts with the initial ϕ, ϕ_0, determined above. This process is:

(1) Perform a maneuver on a copy of the original RSO using the ϕ_0 calculated using the process described earlier in this section
(2) Propagate the RSO copy through the initial maneuver and the maneuver back to its original position, or only until after TCA if it is not maneuvering back to its original position.
(3) For each step of the propagation, run the conjunction detection function, but only against the RSO maneuvering. I.e., check every RSO against the maneuvering RSO instead of checking every RSO against every other RSO.
(4) If a collision is found, increase the maneuver's ϕ value
(5) Return to step one with the increased ϕ value
(6) If no collision is found, use the final ϕ value to run the maneuver on the actual RSO
(7) Propagate the maneuvered RSO forward until it returns to its original orbital position, or until the end of the propagation window if RSOs are not maneuvering back, and save the new data for continuing on with the main conjunction detection process.

While the value by which ϕ changes with every iteration can be adjusted if desired, it is initially set to increase by ϕ_0 every iteration. Testing showed that smaller adjustment values often did not eliminate the new conjunction that was found, requiring multiple iterations to avoid the same conjunction and using additional computational time. By using multiples of ϕ_0, if an iteration i finds a new conjunction, the next iteration $i + 1$ will likely avoid that conjunction. Finally, only 15 iterations are attempted. If a conjunction is found on the 15th iteration the final ϕ is used and the δv is recorded. It is assumed a satellite operator would be able to determine a more precise and unique maneuver to avoid all conjunctions for the same, or less, δv compared to the δv it takes to maneuver by $15\phi_0$. This means that the final δv recorded will be an overestimation of what would likely be the true δv.

CALCULATING PROBABILITY OF COLLISION

Probability of collision is an important metric for conjunctions, and one that operators frequently use to determine whether they should maneuver their satellite to avoid a collision. In calculating

the probability of collision (P_c) for a conjunction, four assumptions or simplifications are made. First, the relative motion between two RSOs during a conjunction is considered linear due to the high relative velocities found in on-orbit. Such high relative velocities mean the relatively small acceleration due to gravity is negligible during the period of conjunction. Second, the positional errors are zero-mean, Gaussian, and uncorrelated. Third, the positional covariance, its shape, size, and orientation, is assumed to be constant due to the relative short period of the conjunctions. Finally, the RSOs are modeled as spheres. Modeling the RSOs as spheres with a radius equal to half the largest dimension of each RSO will overestimate the probability of collision as the sphere model encompasses the entire satellite. The spherical assumption reduces complexity in the calculation of P_c and reduces the need for observation and tracking data on the orientation of objects; data that are often difficult to produce or find.[7]

Covariance Matrix Generation

The first step in calculating the probability of collision is generating the covariance matrices that quantify the positional error for each RSO. This positional error at TCA is comprised of uncertainty from the observation, e.g. uncertainties in the equipment used to measure the position of an RSO, and uncertainty from the propagation of the RSO's state into the future. The uncertainty due to the methods the 18SPCS uses in state determination is not released publicly. Because we use SGP4, the propagation uncertainties can be estimated. The process to do so is explained below.

The covariance C is the expectation of the product of the deviation of variables from their mean. For a vector, the covariance is given by Equation (19).

$$C_v = E[(v - m)(v - m)^T] \tag{19}$$

In Equation (19), m is the vector containing the mean of each element of v. If there were N independent measurements of the state vectors for each RSO (X_i), and the true state were known, the true errors could be calculated. However, because the data given by the 18SPCS is not in the form of position and velocity states, the true state cannot be known. Instead, by running the 18SPCS data through SGP4 at a time from epoch of 0, a best estimate, $\bar{X}$, is determined. With this best estimate, instead of true error, residuals, defined as

$$\delta X_i = X_i - \bar{X} \tag{20}$$

are calculated. In these conditions, the covariance is given by Equation (21).

$$C_v = E[(\delta X_i - m)(\delta X_i - m)^T] \tag{21}$$

In Equation (21), m is again defined as a vector comprised of the mean of each element of the residuals, or $E[\delta X_i]$.

When calculating the covariance matrix, historical data from the 18SPCS is used to provide the multiple state estimates needed to generate the covariance. The period of time from which historical data is taken is equal in length to the period of propagation for which error is being estimated. For example, if the error associated with three days of propagation is needed, the most current data is used for the best estimate. Then all the data between the current data and data from three days ago is stored. This allows the simulation to represent the time before a possible conjunction an operator would be notified of the conjunction. In the previous example, if an operator is notified of a conjunction occurring in three days, the notifying organization must be propagating forward three days in time. The RSOs' state vectors at TCA for this conjunction would thus have an uncertainty associated with three days' worth of propagation. In the simulation, generating the uncertainty

associated with three days of propagation simulates the window between the notification of a conjunction to an operator and the time of the conjunction.

After gathering the data from the 18SPCS, the actual state values must be determined. First, as mentioned before, the most current data available is run through the SGP4 propagator at the data's included epoch to generate the best estimate for the true state. For example, the TLE for a certain RSO might give the state values at noon on January 1st, 2021. So, that RSO's TLE data is run through the SGP4 propagator at noon on January 1st, 2021 to convert the TLE's data format into the position and velocity vectors for the RSO at noon on January 1st. Then, all the historical data is propagated to the same point. In the same example, data from 10:00 AM on December 30th, 2020 would be propagated to noon on January 1st, 2021. Equation (22) is used to calculate residuals, δx_i for each object with X_i being each individual state estimate except for the best estimate, which is $\bar{X}$.

$$\delta x_i = X_i - \bar{X} \tag{22}$$

After generating all the residuals for each RSO, the mean vector, m, of the residuals is calculated for each RSO. The mean vector is a vector where each element is the mean of the corresponding element in the residuals. Equation (23) shows the calculation used to generate m. The covariance is then generated using Eq. (24), where N is the number of state estimates, including the best estimate.[8]

$$m = \frac{\sum_{i=1}^{N-1}(\delta X_{epoch})_i}{N-1} \tag{23}$$

$$C_v = \frac{\sum_{i=1}^{N-1}(\delta X_{epoch} - m)_i \left(\delta X_{epoch} - m\right)_i^T}{N-1} \tag{24}$$

Probability of Collision Evaluation

The next step, after determining the covariance matrices for the two RSOs conjuncting, in producing a collision probability is summing the covariance matrices, represented graphically as error ellipsoids, of the two conjuncting RSOs. This is possible due to the expectation of uncorrelation between the two error ellipsoids. This single larger ellipsoid is centered at one of the two RSOs, now considered the primary object. The other RSO, or secondary object, passes through this ellipsoid rapidly along the relative velocity vector. The spherical nature of the secondary object along the line of relative motion creates a cylindrical path often called a collision tube. Figure 5 shows this collision tube. If the secondary sphere comes within a distance equal to the sum of the two RSOs' radii from the primary object, there is a physical overlap and a collision.

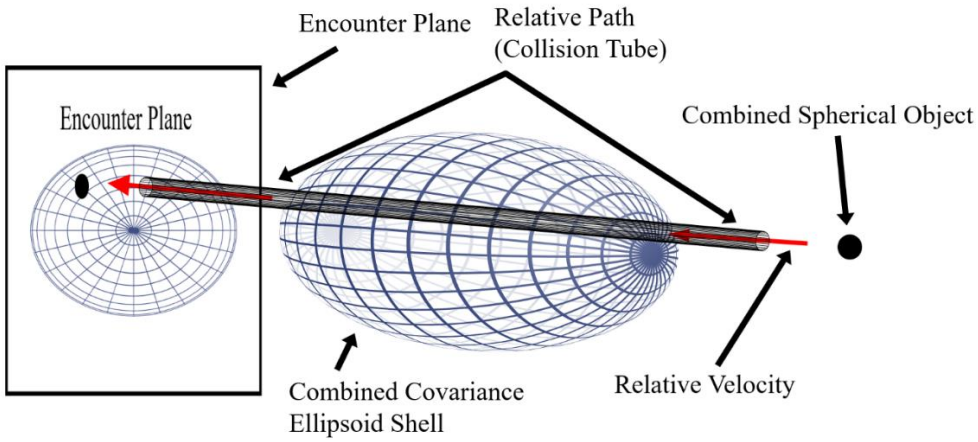

Figure 5. Collision tube passing through combined covariance ellipsoid along relative velocity direction (adapted from [7]).

Evaluating the three dimensional integral across the probability density function (PDF) within the collision tube produces P_c. The linear nature of the collision tube means the dimension along which the relative velocity vector lies can be decoupled. Meaning combined error ellipsoid and collision tube can be projected as an ellipse and circle respectively onto a plane normal to the path of the collision tube. This plane is called the encounter plane, and the projection is shown in Figure 6. The radius of the smaller circle, or combined spherical radius, is equal to the radius of the collision tube which is the sum of the radii of the spherical models of the two RSOs.

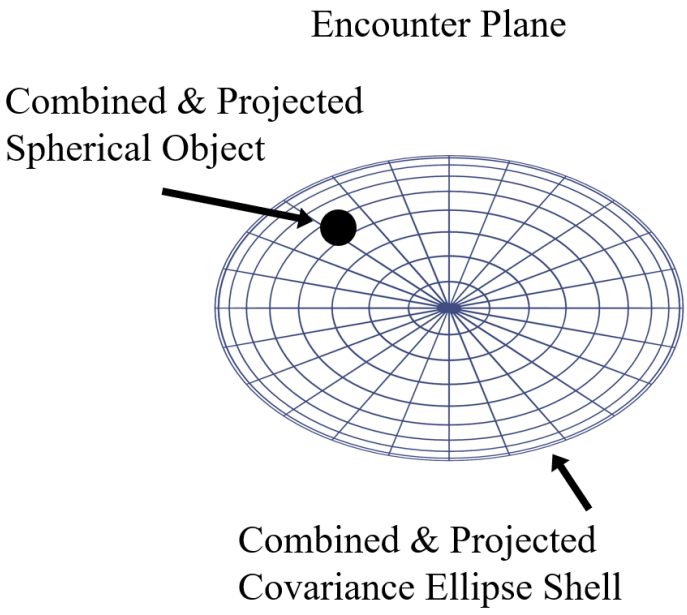

Figure 6. Projection of collision tube and combined covariance ellipsoid onto Encounter Plane, reducing collision probability problem to two-dimensional integral (adapted from [7]).

To perform this projection, all relevant vectors are transformed into a coordinate frame in where the z direction is normal to the encounter frame or aligned with the relative velocity vector. This process first requires applying a rotation to align the z axis with the relative velocity vector. To achieve this projection, the z direction is ignored to transform everything into a two-dimensional problem on the encounter plane.

The rotation matrix for aligning the z direction with the relative velocity vector is given as

$$R_{encounter} = [\hat{\imath};\ \hat{r}; \hat{v}]^T \tag{25}$$

where

$$\hat{r} = \frac{\boldsymbol{r}_0}{\|r_0\|};\ \hat{v} = \frac{\boldsymbol{v}_0}{\|v_0\|};\ \hat{\imath} = \hat{r} \times \hat{v} \tag{26}$$

and $\boldsymbol{r}_0$ and $\boldsymbol{v}_0$ are the relative position and velocity vectors at TCA. This rotation is then applied to the combined covariance matrix. This operation is simply matrix multiplication of three 3 by 3 matrices. Equation (27) shows how to apply a rotation matrix $\boldsymbol{R}$ to a matrix $\boldsymbol{M}$.

$$\boldsymbol{M'} = \boldsymbol{RMR^{-1}} \text{ however, the inverse of a rotation matrix is its transpose, so } \boldsymbol{M'} = \boldsymbol{RMR^T} \tag{27}$$

After rotating the covariance matrix, the third row and column are eliminated to give a two-dimensional matrix representing the error ellipse in the encounter plane. When applying this transformation to the relative position vector, due to the definition of the rotation, the vector at TCA becomes a vector of the same magnitude solely along the y direction

This projection reduces the dimensionality of the PDF integral from three to two. Now, P_c is found by integrating the PDF of the error ellipse within the smaller circle that is the projection of the collision tube. This integral is evaluated at the time of closest approach (TCA). Equation (28) shows this two-dimensional integral in cartesian space.

$$P_c = \frac{1}{2\pi\sigma_x\sigma_y}\int_{-OBJ}^{OBJ}\int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} e^{-\frac{1}{2}\left[\left(\frac{x-x_m}{\sigma_x}\right)^2+\left(\frac{y-y_m}{\sigma_y}\right)^2\right]} dy\, dx \tag{28}$$

In this equation, x lies along the minor axis of the covariance ellipse, y lies along the major axis of the covariance ellipse, σ_x is the standard deviation of the positional uncertainty in the x direction, σ_y is the standard deviation in the y direction, OBJ is the combined spherical object's radius, and x_m and y_m are the components of the relative position vector at TCA in the x and y directions respectively.

To find σ_x, σ_y, x_m, and y_m, the covariance must be diagonalized. In its diagonal form, σ_x is the smaller of the two values on the diagonal and σ_y is the larger one. Because the covariance matrix will always be two-dimensional, the diagonalization will not require much computation. The transformation matrix, composed of the eigenvectors of the covariance matrix is then used to transform the relative position vector to the diagonal basis. Equation (29) shows this process.

$$\begin{aligned} C_{v,diagonal\ frame} &= T^{-1}C_vT \\ r_{diagonal\ frame} &= T^{-1}r_{encounter\ plane} \end{aligned} \tag{29}$$

However, in this case, T^{-1} can be substituted for T^T because the inverse of an orthogonal matrix is equal to its transpose, the eigenvalues of a symmetric matrix are orthogonal, and the covariance matrix will always be symmetric. This helps the collision probability computation run faster as the transpose of a matrix is trivial to find, while its inverse can take much more effort depending on the algorithm used, even for a two-dimensional square matrix.

After transforming the variables in the P_c integral to the correct form, the integral must be evaluated. First, another simplification is made to make evaluation of the integral easier. Using the error function, defined in Equation (30), the integral can be simplified to the single integral given in Equation (31).

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{30}$$

$$P_c = \frac{1}{\sqrt{8\pi}\sigma_x} \int_{-r}^{r} \left[\operatorname{erf}\left(\frac{y_m + \sqrt{r^2 - x^2}}{\sqrt{2}\sigma_y} \right) + \operatorname{erf}\left(\frac{-y_m + \sqrt{r^2 - x^2}}{\sqrt{2}\sigma_y} \right) \right] e^{\frac{-(x+x_m)^2}{2\sigma_x^2}} dx \tag{31}$$

To evaluate this integral, a version of Simpson's 1/3 rule is used, developed by Alfano.[9] First the odd-order terms are calculated using Equation (32), and the even-order terms are given in Equation (33). Finally, because testing shows Simpson's rule underestimates these integrals at the limits, a zeroth-order adjustment is used, given in Equation (34).

$$y(x) = \sqrt{r^2 - x^2} \text{ and } dx = \frac{r}{2m}$$

$$k_{odd} = 4 \sum_{i=1}^{k} \left[\operatorname{erf}\left(\frac{y_m + y(x)}{\sigma_y \sqrt{2}} \right) - \operatorname{erf}\left(\frac{y_m - y(x)}{\sigma_y \sqrt{2}} \right) \right] \left[e^{\frac{-(x_m+x)^2}{2\sigma_x^2}} + e^{\frac{-(x_m-x)^2}{2\sigma_x^2}} \right] \tag{32}$$

$$\text{Where } x = (2i - 1) * dx - r$$

$$k_{even} = 2 \sum_{i=1}^{k-1} \left[\operatorname{erf}\left(\frac{y_m + y(x)}{\sigma_y \sqrt{2}} \right) - \operatorname{erf}\left(\frac{y_m - y(x)}{\sigma_y \sqrt{2}} \right) \right] \left[e^{\frac{-(x_m+x)^2}{2\sigma_x^2}} + e^{\frac{-(x_m-x)^2}{2\sigma_x^2}} \right] + 2 \left[\operatorname{erf}\left(\frac{y_m + r}{\sigma_y \sqrt{2}} \right) - \operatorname{erf}\left(\frac{y_m - r}{\sigma_y \sqrt{2}} \right) \right] e^{\frac{-x_m^2}{2\sigma_x^2}} \tag{33}$$

$$\text{Where } x = 2i * dx - r$$

$$k_0 = 2 \left[\left[\operatorname{erf}\left(\frac{y_m + y(x)}{\sigma_y \sqrt{2}} \right) - \operatorname{erf}\left(\frac{y_m - y(x)}{\sigma_y \sqrt{2}} \right) \right] \left[e^{\frac{-(x_m+x)^2}{2\sigma_x^2}} + e^{\frac{-(x_m-x)^2}{2\sigma_x^2}} \right] \right] \tag{34}$$

$$\text{Where } x = 0.015dx - r$$

P_c is then found by combining the expressions together using Equation (35). Additionally, numerical testing shows an optimal k for speed and accuracy can be determined using Equation (36), with a lower limit of 10 and an upper limit of 50.

$$P_c = \frac{dx}{3\sqrt{8\pi}\sigma_x} (k_0 + k_{even} + k_{odd}) \tag{35}$$

$$k = \operatorname{int}\left(\frac{5r}{\min\left(\sigma_x, \sigma_y, \sqrt{x_m^2 + y_m^2} \right)} \right) \tag{36}$$

SIMULATION FLOW

The flow of the simulation is as follows:

(1) Reads in the simulation input file. This file sets up the simulation. It includes data such as propagation window, propagation time step, options for conjunction detection and maneuvering, gives the file paths and the file names of the other input files, etc.
(2) Reads in TLE ephemerides.
(3) Reads in additional data files such as the Union of Concerned Scientists' Satellite Database (UCSSD) that lists attributes of active satellites, the NORAD Satellite Situation Report containing data on RSOs like country of origin, the European Space Agency DISCOS database containing physical dimensions of RSOs, etc. These attributes are saved for each RSO when available or updated.
(4) Propagates all satellites in TLE input file through given propagation window and saves data to file.
(5) If user requests, covariance matrix for each RSO is generated using the historical TLE ephemerides file.
(6) If user requests conjunction detection and maneuvering, simulation begins looping through each time step in the propagation window.

For each time step:

(7) State of each satellite is read in, and conjunction detection is run. If a conjunction is detected, the associated collision probability is calculated within the same function.
(8) If a conjunction is detected, and maneuvering is possible,[*] the RSO(s) to maneuver is/are determined based on the maneuvering rules defined by the user, e.g. if there are two active satellites, the user may want only one satellite to maneuver, or he may want both to maneuver half of the distance in opposite directions.
(9) The sufficing routine is run to determine the magnitude of the maneuver, or maneuvers, that do not cause additional conjunctions.
(10) A maneuver, or maneuvers, with the magnitude(s) given by the sufficing function is implemented, and the new data replaces the old data in the file from the propagation step.
(11) Steps 7 through 10 are repeated for each time step in the propagation window.
(12) After all time steps have been analyzed, the conjunctions found, along with data associated with each conjunction, and the δv each RSO expended maneuvering throughout the time window are saved to output files.

This information is also represented below in Figure 7 as a flow chart.

[*] For a maneuver to be possible, both the user must request that RSOs maneuver out of the way of conjunctions when possible in the setup file, and at least one of the RSOs involved in the conjunction must fit the user's definition of an RSO that is allowed or able to maneuver.

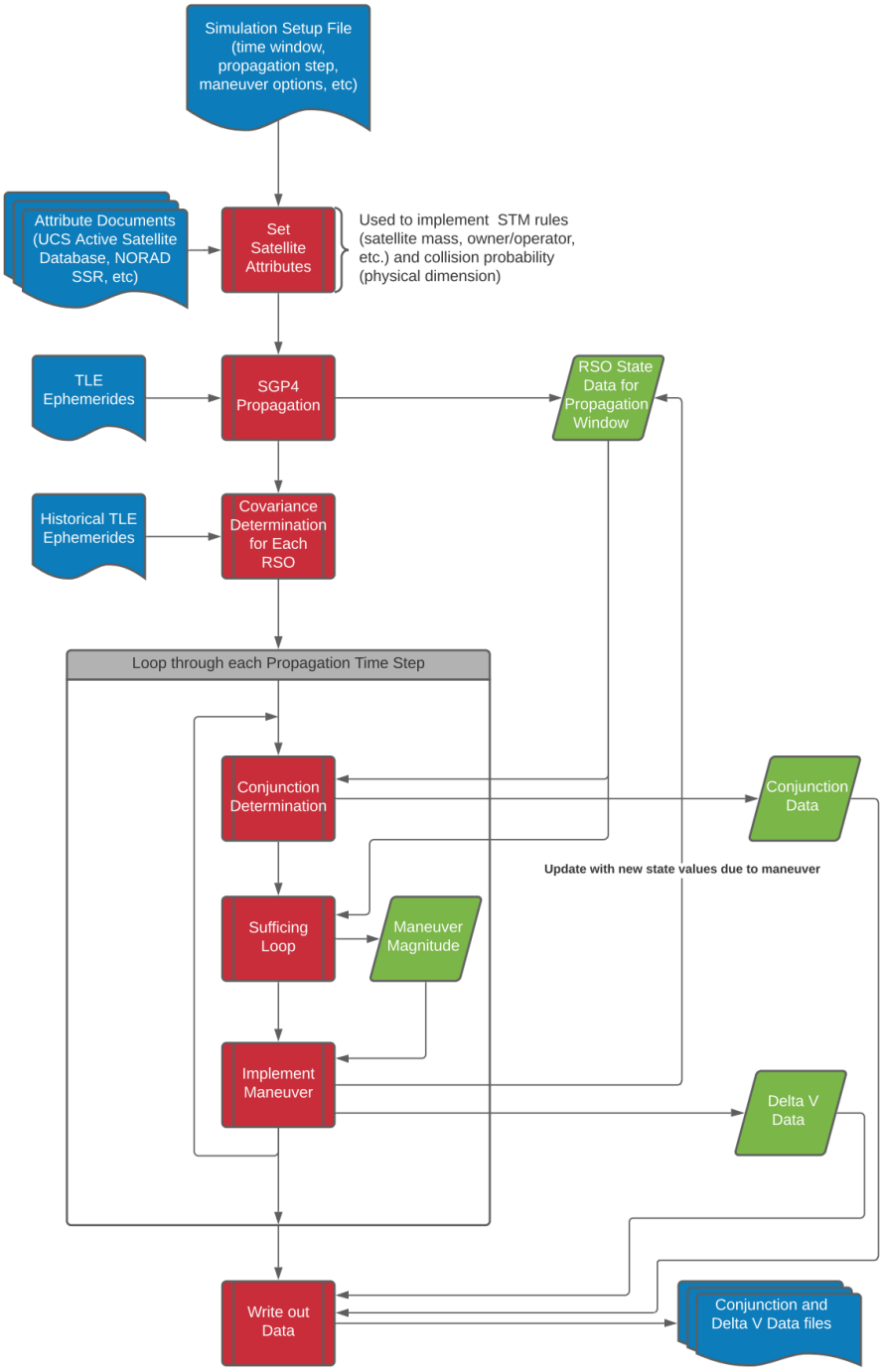

Figure 7. Simulation Flow Chart.

VERIFICATION

Every aspect of the simulation, excluding SGP4 because its code was used without modification, was tested to verify it performed and produced results as expected.

Conjunction Analysis

To verify the simulation, the conjunction determination function was first checked against the Center for Space Standards and Innovation's (CSSI) Satellite Orbital Conjunction Reports Assessing Threatening Encounters in Space (SOCRATES).[*] SOCRATES is a database providing information on possible conjunctions in orbit that looks forward approximately one week. It catalogs conjunctions with an R_{cr} of 5.0 kilometers. SOCRATES' online portal allows for searching its database for conjunctions involving specific RSOs using their NORAD catalog numbers or their names, but it does not offer the ability to obtain a copy of the entire database. So, a sample of conjunction found by the conjunction determination function had to be verified manually against SOCRATES. To do so, 120 random samples of the conjunctions determined by a simulation were chosen, and then the NORAD IDs of the two RSOs involved were entered into the SOCRATES portal. The TCA, range at TCA, and relative velocity at TCA were compared to that given by the simulation to ensure they matched. While this process verifies that the simulation is not overcounting conjunctions, it does not provide any data on whether the program is undercounting conjunctions. The number of conjunctions with an R_{cr} of 1.0 km, in one month, involving active satellites, using TLEs from November 3rd, was determined to be 4,228; similar to the estimated 4,000 according to the Space Data Center.[10]

Additionally, the conjunction determination function was tested against artificial data. This required generating artificial TLEs known to produce a conjunction. For this, a program was written that takes input TLEs and produces output TLEs for fictional RSOs that will be at the same position as the RSOs of the input TLEs at a given time. The sets of RSOs with the same position were then given different velocity vectors to simulate most real conjunctions involving high relative velocities. This was done by propagating the input TLEs to the user-defined time, then producing an RSO with the same position, but a different velocity vector. This position and velocity state were converted to mean orbital elements using the method described above, and then formatted as a TLE. 120 TLEs were chosen randomly from the set of LEO TLEs provided by the 18SPCS. These TLEs were run through the program to produce 120 additional TLEs that should produce a conjunction with the original TLEs. The two sets of TLEs were combined and run through the conjunction determination function to ensure it detected each of the conjunctions.

Implementing Maneuvers

Next, to test the maneuvering routing was functioning correctly, the algorithm to determine mean elements was tested. The aforementioned program also involved a self-checking mechanism. After it produced an artificial TLE, it would read in that TLE and propagate the RSO to the given time to ensure the position and velocity matched the desired position and velocity that was input into the mean element algorithm.

The second part to test was that the implementation of an initial maneuver, the one with ϕ_0 as its phasing angle, was sufficient to avoid the conjunction the maneuver was intended to avoid. This simply required running a conjunction check at the TCA of the original conjunction between the two RSOs after at least one of them maneuvered to ensure they no longer were within the R_{cr} or

[*] https://celestrak.com/SOCRATES/

produced a conjunction with a probability of collision ≥ the minimum probability defined in the user input.

Finally, the sufficing routine needed to be verified. Because the maneuvers implemented by the simulation are not performed by any real satellite operators, artificial data needed to be used. To do this, the program from the conjunction analysis section was utilized again. First, it produced TLEs that would result in a conjunction with the given TLEs like before. Then this data was run through the simulation to implement the initial maneuvers to avoid the first-level conjunctions. The state data of the maneuvered satellites was input into the same program to produce another TLE for an RSO that would conjunct with one of the maneuvering satellites while it maneuvered. Then the data including the two artificial TLEs for every real TLE was run through the simulation again to determine if the sufficing check resulted in a maneuver that avoided both artificial conjunctions.

Collision Probability

Because there is no large database that uses the same method as the one described above to calculate probability of collision for a large number of real conjunctions, another random sample of conjunctions was manually tested. For each sample, the entire process was checked by hand to ensure the covariance matrices for the two RSOs were correct and the estimate of the two-dimensional integral matched the results given by MATLAB's "integral2" double numerical integrator with a relative tolerance of 1E-10.

INITIAL RESULTS

Below are some initial results produced by runs of the simulation for different STM test cases.

Worst-case Scenario

The first simulation cases consisted of a set of runs to show the worst possible scenario. In this case none of the RSOs maneuver to avoid conjunctions. This is worse than the current environment on-orbit as some operators do choose to avoid possible collisions.[11] However, there is no requirement to maneuver under current policy. Because operators decide if and when to maneuver to avoid conjunctions, there is no real-life baseline to which the simulation can be compared. This worst-case scenario, or the scenario in which no satellites maneuver to avoid conjunctions, is used as a baseline since there are currently no clear guidelines as to how and when an operator must maneuver.

The worst-case scenario was run with a R_{cr} of 5.0 kilometers. The TLE ephemerides were obtained from the 18SPCS on April 7th, 2021. Only RSOs in LEO as defined by the 18SPCS, an RSO with a mean motion greater than 11.25 revolutions per day and eccentricity less than 0.25, were provided as inputs to the simulation. The propagation window was a period of thirty days starting on March 22nd, 2021 and ending on April 21st, 2021 with a time step of 2 minutes. Table 1 shows the results for this scenario.

Table 1. Results from Worst-Case Scenario Run.

Total Number of Conjunctions	427,502
Average Probability of Collision across the Conjunctions	5.65E-09

Best-Case Scenario

The next scenario tested was the best possible scenario. This included a case in which every single conjunction involving at least one active satellite, as defined by the UCSSD, maneuvered to avoid those conjunctions. Table 2 shows the results from this scenario. For this case, the simulation parameters were the same as mentioned above. There were some additional parameters, however. The notification window, or the propagation time used to calculate the covariance matrices, was 3 days, and every maneuver was initiated 2 days before the TCA.

Table 2. Result from Best-Case Scenario Run.

Total Number of Conjunctions	427,698
Average Probability of Collision across the Conjunctions	5.70E-09
Number of Conjunctions Involving at least 1 Active Satellite	86,542
Average Probability of Collision for Conjunctions involving at least 1 Active Satellite	2.72E-06
Average Delta V Required for Satellites involved in at least 1 Conjunction	5.25E-05 kilometers per second

STM Test Case

Because university satellites are usually small, lack any propulsion system, and are usually owned by a non-profit entity, it is assumed these RSOs will be exempt from maneuver requirements. For this scenario, satellites defined as university or institute owned were considered non-active satellites. This case was used to compare total conjunctions with and without university satellites to determine realistic expectations of future operators.

The input parameters for the simulation are the same 1-month propagation window as above with the same propagation time step and the same input ephemerides. The R_{cr} was set to 2.5 kilometers with a notification window of 3 days and a maneuver time of 2 days before the conjunction. Table 3 below shows the results for this scenario.

Table 3. Result from Best-Case Scenario Run.

Total Number of Conjunctions	63526
Average Probability of Collision across the Conjunctions	2.11E-06
Number of Conjunctions Involving at least 1 Active Satellite	3,957
Average Probability of Collision for Conjunctions involving at least 1 Active Satellite	4.66E-06
Average Delta V Required for Satellites involved in at least 1 Conjunction	2.03E-04 kilometers per second

CONCLUSION

In the effort to develop possible STM frameworks to help preserve the space environment, a method of rigorously testing the implications of such a framework is necessary. This work discusses development of a simulation to assist in the analysis of proposed STM frameworks. To be useful in this effort, the simulation includes a number of features required to produce the data needed. It combines a method of determining the state of RSOs at any time, a method to determine conjunctions between two RSOs, which can be defined by either the range of closest-approach or probability of collision, and a method to maneuver RSOs out of harm's way. The combination of these features allows for the collection of data such as: the number of conjunctions before the implementation of proposed STM policies, features of those conjunctions like their TCA, range at TCA, relative velocity at TCA, the number of conjunctions that a proposed STM regime would prevent, the δv each RSO capable of maneuvering would need to expend to maintain compliance with an STM policy of interest, and more. Additionally, this simulation has the flexibility to test future space environments with many more RSOs as more actors become involved in space and large constellations of satellites are developed and deployed. Finally, the simulation implements a framework with tools for the implementation of unique and inventive STM policy proposals.

ACKNOWLEDGEMENTS

We would like to thank the Georgia Tech 2020 Small Bets Seed Grant for providing funding for this research. We would also like to thank the Georgia Tech Partnership for an Advanced Computing Environment (PACE) for the computing resources that allowed us to run and test our tool.

REFERENCES

[1] Jenkin, A., Sorge, M., Peterson, G., McVey, J., & Yoo, B. (2015). Predicting the Future Space Debris Environment. Crosslink, 16(1), 8-13.

[2] Muelhaupt, Theodore & Sorge, Marlon & Morin, Jamie & Wilson, Robert. (2019). Space traffic management in the new space era. Journal of Space Safety Engineering. 6. 10.1016/j.jsse.2019.05.007.

[3] Vallado, David & Crawford, Paul & Hujsak, Richard & Kelso, T.S.. (2006). Revisiting Spacetrack Report# 3: Rev.

[4] Aida, Saika & Kirschner, Michael. (2013). Accuracy Assessment of SGP4 Orbit Information Conversion into Osculating Elements.

[5] Rodríguez, J. & Martínez, Francisco & Klinkrad, H.. (2002). Collision Risk Assessment with a `Smart Sieve' Method. 486. 159.

[6] Andersen, D. E. (1994). Computing NORAD Mean Orbital Elements From A State Vector. AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH.

7 Alfano, S., & Oltrogge, D. (2018). Probability of collision: Valuation, variability, visualization, and validity. Acta Astronautica, 148, 301-316. doi:10.1016/j.actaastro.2018.04.023

[8] Osweiler, Victor. (2006). Covariance Estimation and Autocorrelation of NORAD Two-Line Element Sets. 130.

[9] Alfano, Salvatore. (2007). Review of conjunction probability methods for short-term encounters. Advances in the Astronautical Sciences. 127. 719-746.

10 Foust, J. (2020, November 03). Space traffic management idling in first gear. Retrieved from https://spacenews.com/space-traffic-management-idling-in-first-gear/

[11] Alfano, Salvatore & Oltrogge, Daniel & Krag, Holger & Merz, Klaus & Hall, Robert. (2021). Risk assessment of recent high-interest conjunctions. Acta Astronautica. 184. 10.1016/j.actaastro.2021.04.009.