MAE 6760: Model-Based Estimation

Collision Detection between Objects in Orbit

via Alfano's Method with an Extended Kalman Filter (EKF)

Eric Xue[1]

[1]Cornell University, Masters of Engineering Student

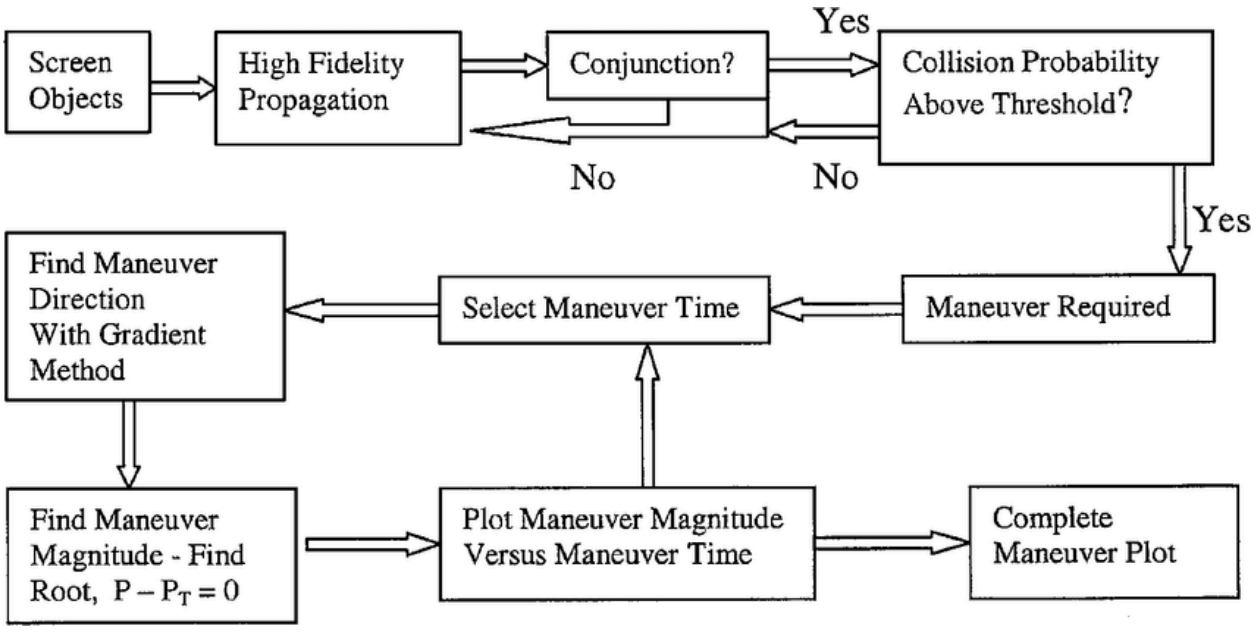
May 17, 2024

Nomenclature

r_p, r_s	*Position of Primary / Secondary Object*	$\circ^I$	*Superscript Denotes Inertial Frame*
r	*Relative Position between Objects*	$\circ^E$	*Superscript Denotes Encounter Frame*
v_p, v_s	*Velocity of Primary / Secondary Object*	$\circ^P$	*Superscript Denotes Principle Frame*
v	*Relative Position between Objects*	$\Delta t, dt$	*The time step for linearization*
R_{obj}	*Combined Spherical Radius*	x, x_k	*State vector at stage k*
Cov	*Covariance Matrix*	Z_{meas}	*Sensor Measurements*
σ_x, σ_y	*Standard Deviation of x and y in Principle Frame*	w	*Random Disturbance to the system*
P_C	*Probability of Collision*	v	*Random Sensor Noise*
$x_m y_m z_m$	*Components of relative Position.*	**F, G, H**	*Linearized system matrices*
$e_x e_y e_z$	*Axis defining Encounter Frame*	P_k	*Covariances of state at stage k*
R_{X2Y}	*Rotation matrix from FrameX to Y*	Q, R	*Covariances of disturbance & sensor noise*
n	*The upper limit for summation*	μ	*Gravitational parameter of Earth*

1. Introduction

Spacecraft collision avoidance encompasses the execution and investigation of procedures aimed at mitigating the probability of spacecraft inadvertently colliding with other objects in orbit. With space becoming more crowded with mega constellations, debris, commercial space shuttles, and more, it becomes increasingly pressing for operators of these spacecraft to have a well-conceived probability of an incoming collision and whether it falls under a certain acceptable risk. This becomes especially true when the majority of man-made objects in space are operating in Low Earth Orbit (LEO), where orbital speeds reach several kilometers per second, resulting in a significant amount of kinetic energy and thus increased destruction upon impact, while reducing the amount of time to react to such a collision.



This project will focus on the detection portion of this problem, more specifically, the probability calculation of an oncoming collision. This can be decoupled from the maneuvering of the spacecraft once the probability of an oncoming collision surpasses a certain threshold. While the work can be scaled up to track multiple objects, the primary spacecraft of focus will be the International Space Station (ISS) due to its residing in LEO with its location readily available to the public, and its impact in advancing scientific research from its inception. A second spacecraft with a similar but intersecting orbit will be observed, resulting in two places within an orbital period of the ISS where a spike in the probability should occur. In reality, the secondary object should be small debris, but another spacecraft has more readily available options of being tracked when it is cooperative. It will be assumed that both spacecraft will be equipped with GPS receivers, obtaining the measured position vectors of each directly as opposed to range and angle measurements from an observatory.

An Extended Kalman Filter (EKF) will be used to propagate the error covariances of the state. At first, the update step was ignored to simulate propagating the state/covariance forward to predict a probability of collision with an oncoming spacecraft and to test the calculation. Next, a more realistic case of the EKF updating the state while simultaneously running calculations of collision, resulting in a more accurate probability as the states and covariance are updated.

2. Theory

Before a probabilistic model can be determined, several simplifying assumptions regarding an encounter between two objects in space must be made. Normally, the probability of a collision relies on the integration of a probability density function (PDF) in 3-D. When performing this integration over this domain, it becomes almost impossible to do so analytically and has no closed-form formula even when assuming a Gaussian distribution [1]. To get around this, a simplifying model called the Short-Term Encounter (STE) model is devised. This model focuses on the Time of Closest Approach (TCA), the point in time when the two objects' distances are at a minimum. This is also the point in which the probability is likely to be the highest and when the duration of the encounter is short. At TCA, the model also assumes the magnitude of the relative velocity is sufficiently large compared to the relative acceleration, which fortunately describes the majority of encounters in LEO.

Due to the large magnitude of the relative velocity, the relative position between the two objects is assumed to be rectilinear, meaning the second object will pass by the primary object in a straight line. It is also assumed that the velocities of the objects at the TCA are deterministic for a reference frame to be developed. Generally, this does not carry over to the positions as the mean is taken for calculations, but for the sake of this project, they will also be deterministic. This is a fair assumption as the orbital dynamics governing a 2-body system will be accurate enough during the prediction step and there arent large disturbances affecting the orbit. This assumption doesn't impact any future calculations nor the generalization of this work as it is trading the mean value position for the true position, which given enough measurements should theoretically be the same. Lastly, the objects of interest will be assumed to be spherical in nature and defined by their respective radius as is common with this probabilistic analysis of collision.

2.1 Encounter Model / Frame

With the motion being rectilinear, this motivates a new frame where one of the axes is aligned along the relative velocity. One option defines the Encounter Frame, which sees the origin centered at the position of the primary object r_p and is derived from the Encounter plane. This plane intersects the origin and is orthogonal to the relative velocity ($v = v_s - v_p$). Thus, e_z will be defined as being along the relative velocity, e_x pointing towards the projection of the relative position ($r = r_s - r_p$) onto the encounter plane, and e_y is the orthogonal axis that completes this frame [1].

$$e_z = \frac{v}{||v||} \qquad e_y = \frac{v \times r}{||v \times r||} \qquad e_x = e_y \times e_z \tag{Eq.1}$$

From this formulation, it can be seen that the relative position vector lines on the $e_z - e_x$ plane ($r \perp e_y$). Thus, one can project the relative position vector onto the two axes via the dot product to obtain the vector in components of this frame (Eq. 2):

$$x_m^E = r \bullet e_x \qquad y_m^E = r \bullet e_y = 0 \qquad z_m^E = r \bullet e_z \tag{Eq.2}$$

Note that the basis vectors of the encounter frame also form the rotation matrix from the inertial frame (I) to the encounter frame (E), R_{I2E} (Eq.3):

$$R_{I2E} = [e_x, e_y, e_z] \quad \text{(Eq.3)}$$

such that one can obtain the covariance matrix in the E-frame from the I-frame via the following (Eq.4):

$$Cov^E = R_{I2E} * Cov^I * (R_{I2E})^T \quad \text{(Eq.3)}$$

Lastly, the rotated covariance ellipse must be diagonal for further simplifications of the integrand for the probability calculation in later sections [1]. In the case that the covariance matrix isn't diagonal, an additional transformation must be performed. A matrix is diagonalizable via its eigenvalues/vectors, where the matrix formed by the eigenvectors,R_{E2P}, is the rotation into this principle axis frame (P), and the resulting diagonal covariance is simply the diagonal matrix formed by the respective eigenvalues. This is readily implementable via MATLAB's diag() function (Eq.4). Note that this step should be done after it is confirmed that the covariance isn't diagonal to begin with, or else diag() will return a nonsensical rotation matrix.

$$\begin{gathered} [V, D] = diag(Cov^E) \\ R_{E2P} = V \\ Cov^P = D = R_{E2P} * Cov^E * (R_{E2P})^T \end{gathered} \quad \text{(Eq.4)}$$

The rotated position in this new frame is as follows (Eq.5):

$$r_m^P = [x_m^P; y_m^P; z_m^P] = R_{E2P} * r_m^E = R_{E2P} * [x_m^E; y_m^E; z_m^E] \quad \text{(Eq.5)}$$

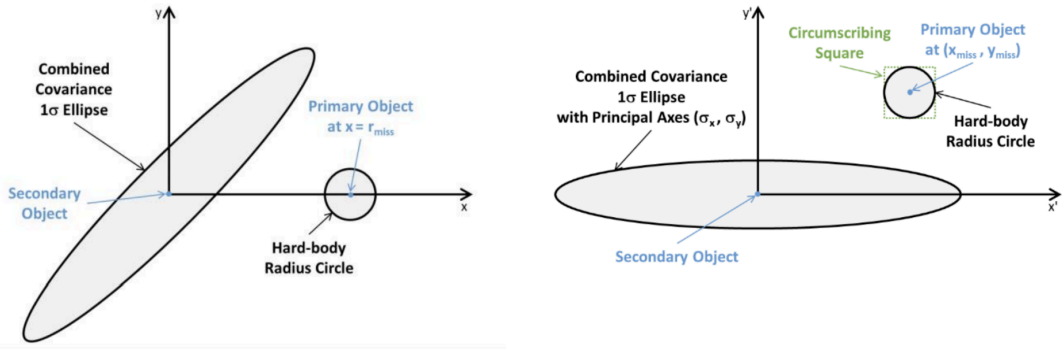


Figure 1: Visualization of Encounter Frame Axis vs Principle Frame Axis [2]

2.2 General Collision Probability Calculation

With the short-term encounter model described, a few more assumptions must be laid out before a general probability calculation can be considered. Firstly, by assuming the covariance of the objects are uncorrelated, they can be summed into a combined covariance ellipsoid centered at the primary object. Under the STE model, due to the relative velocity being dominant and motion rectilinear, the second object will pass through this combined covariance ellipsoid quickly, forming a uniform cylindrical region in the direction of the relative

velocity [3]. Furthermore, due to the spherical object assumption, the radius of each can also be summed to form a combined radius, R_{obj}, as demonstrated by the figure below (Fig.2).

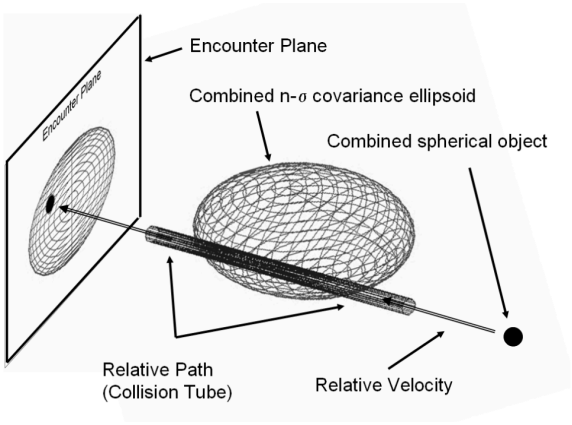


Figure 2: General Overview of Encounter between 2 Objects [3]

The dimension associated with the tube length is decoupled (due to defining the axis along the relative velocity), allowing for the projection of the covariance onto the encounter plane to be an ellipse, while the combined spherical object to shown as a circle [3]. This decoupling collapses the 3-D probability onto a 2-D one , where the resulting PDF on the encounter plane is as follows (Eq.6) :

$$P_C = \frac{1}{2\pi\sigma_x\sigma_y}\int_{-R_{obj}}^{R_{obj}}\int_{-\sqrt{R_{obj}^2-x^2}}^{\sqrt{R_{obj}^2-x^2}} exp\left(-\frac{1}{2}\left[\left(\frac{x-x_m^P}{\sigma_x}\right)^2+\left(\frac{y-y_m^P}{\sigma_y}\right)^2\right]\right)dydx \quad \text{(Eq.6)}$$

From here, this double integral is still not easily solvable, but there exists both numerical and analytical methods that depending on the author, take further approximations to make this solvable. The implementation of choice for this project will be from Alfano due to his method having relatively quick computational time and ease of implementation.

2.3 Alfano's Method

Alfano's method involves breaking the probability integral into a series utilizing the error function, which MATLAB has readily available as erf(). Unfortunately, while multiple resources online reference the original 2005 paper by Alfano, the theory and derivation behind the method are behind a paywall, with only the results being available. Multiple different sources have been reviewed to ensure a holistic understanding of the variables that go behind the following equation and to ensure that the understanding/explanations of the previous sections make reasonable sense in the context of Alfano's method. It is assumed that Alfano's definition of the encounter frame follows the convention shown in this project alongside other papers, providing the notations/definitions used in this report [3]. His infinite series solution is
as follows (Eq.7):

$$P_C = \frac{2R_{obj}}{\sqrt{8\pi}\sigma_x n}\sum_{i=0}^{n}\left[erf\left(\frac{y_m^P + \frac{2R_{obj}}{n}\sqrt{(n-i)*i}}{\sigma_y\sqrt{2}}\right) + erf\left(\frac{-y_m^P + \frac{2R_{obj}}{n}\sqrt{(n-i)*i}}{\sigma_y\sqrt{2}}\right)\right] * exp\left(\frac{-\left(\frac{R_{obj}(2*i-n)}{n}\right)^2 + x_m^P}{2\sigma_x^2}\right) \quad \text{(Eq.7)}$$

The only uncertainty lies in Alfano further breaking up this series into its even and odd components, later defining an upper and lower limit on the number of terms to break it up into [3]. Through this, Alfano defines an upper limit on the number of terms to be 50 for an acceptable level of tolerance. This number does not correlate to a max number for n, and the upper limit for the infinite series must be found via experimentation and seeing when the probability values start to coverage.

3. Simulation Setup

All random variables (state, disturbance, sensor noise) will be assumed to be Gaussian and uncorrelated with each other. The disturbance and sensor noise will also be assumed to be white and zero mean.

3.1 Dynamics

The state that will be tracked will be the 3 components of position and velocity in the inertial frame, with the state being a random Gaussian variable with a mean $\hat{x}_{k|k}$ and covariance matrix P_k:

$$[\mathbf{x}] = \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} \quad \mathbf{x}_k \sim \mathcal{N}(\hat{\mathbf{x}}_{k|k}, P_k^-) \quad \text{(Eq.11)}$$

To simulate the true states and propagate the initial conditions, it will be assumed that the spacecraft is operating under a 2-body orbit with Earth as its central body. The initial states of each spacecraft will be propagated using MATLAB's ode113(), with the following function defining the dynamics:

$$[\dot{\mathbf{x}}] = \begin{bmatrix} v_x \\ v_y \\ v_z \\ a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \mathbf{x}(4) \\ \mathbf{x}(5) \\ \mathbf{x}(6) \\ \frac{-\mu*\mathbf{x}(1)}{\|\mathbf{x}(1:3)\|} + w_x \\ \frac{-\mu*\mathbf{x}(2)}{\|\mathbf{x}(1:3)\|} + w_y \\ \frac{-\mu*\mathbf{x}(3)}{\|\mathbf{x}(1:3)\|} + w_z \end{bmatrix} = f(\mathbf{x}, \boldsymbol{w}) \qquad \boldsymbol{w}_k \sim \mathcal{N}(0, Q) \quad \text{(Eq.12)}$$

Where *w* represents the disturbances in the associated directions with a covariance of Q.

3.2 Sensor Model

As mentioned, this project will look at the cooperative encounter case, where information from the secondary object is known via GPS. While this can be expanded into the non-cooperative scenario, with the secondary object's states being estimated via a sensor that isn't GPS (such as range, latitude, and

longitude measurements from an observatory), that scenario isn't explored. It is assumed that the results will generally be the same with a larger error covariance as a result of the observability matrix being unobservable at certain angle measurements, and would require multiple observatories to mitigate this. The following represents this GPS measurement model:

$$\left[\boldsymbol{Z}_{meas}\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \left[\mathbf{x}\right] + \begin{bmatrix} \mathrm{v}_x \\ \mathrm{v}_y \\ \mathrm{v}_z \end{bmatrix} = H\mathbf{x} + \mathbf{v} \qquad \mathbf{v}_k \sim \mathcal{N}(0,\, R) \tag{Eq.12}$$

where v represents the sensor noise from the GPS.

3.3 EKF / Covariance Propagation

Since the dynamics for a spacecraft in orbit are highly non-linear, an EKF was used to estimate the state and propagate the covariance. The state was predicted using ode113() while the covariance prediction utilizes the linearized system matrices following an Euler Linearization:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \Delta t * \dot{\mathbf{x}}|_k = f_{linear}(\mathbf{x}, \boldsymbol{w}) \\ P^{+}_{predict} &= F * P^{-}_{predict} * F^{T} + G * Q * G^{T} \end{aligned} \tag{Eq.13}$$

where F represents the Jacobian of the Euler linearization of the dynamics of the state, and G represents the same linearizations of the disturbances:

$$F = \frac{\partial f_{linear}}{\partial \mathbf{x}} = \begin{bmatrix} 1 & 0 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 & dt \\ -\frac{\mu}{R^3} & 0 & 0 & 1 & 0 & 0 \\ 0 & -\frac{\mu}{R^3} & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{\mu}{R^3} & 0 & 0 & 1 \end{bmatrix} \qquad G = \frac{\partial f_{linear}}{\partial \boldsymbol{w}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ dt & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & dt \end{bmatrix} \tag{Eq.14}$$

From here, the rest of the steps for the EKF (Kalman Gain calculation, update step), will follow the traditional EKF format. The Δt used for linearization will be proportional to the orbital period of the primary object to save on computational power.

4. Results

As mentioned, the STE model holds only at the time of closest approach. This means the probability calculation will not guarantee any accuracy to the truth when using the equation throughout the entire orbit. Thus, there will be three main simulations done. The first will just test the probability calculation at the time of the closest approach to verify the probability calculation in the way it was intended to be used. This involves using just the predicted state and covariance, removing the update step of the EKF. The second will use the full-time history to calculate the probability at each time step, ignoring the close encounter assumption. This is due to the erf() function returning an extremely low number when the argument is large, which will be the case when the relative motion between the objects is large (anywhere in the orbit far from TCA). This allows for a pseudo-tracking of the probability of collision throughout the orbit, where the probability should have peaks when the distance between the objects is at a minimum and 0 when elsewhere.

4.1 Simulation Constants

The covariance of the disturbance is based on the expectation that any perturbations will be dwarfed by the central body motion of the Earth, and thus should be several orders of magnitude smaller than the velocity. The covariance of the sensor noise is based on GPS having an error in the range of centimeters but magnified for this project. The length of the ISS is about 0.109 km, with an assumed small debris (being treated as a spacecraft for trackability) being 0.001km. Distance units will be in terms of km and velocity units in terms of km/s.

Table 1: Initial Conditions and Constants

x_0^{ISS}	x_0^{obj1}	P_0	Q	R	R_{obj}
[3694.497949; −5254.82323; 2163.0537840; 5.17192942; 1.3753274; −5.49250341]	[6448.969113; −1775.15773; −1081.056219; −1.51966471; −7.07360366; 2.55029041]	diag([0.01^2; 0.01^2; 0.01^2; 0.01^2; 0.01^2; 0.01^2])	diag([0.001^2; 0.001^2; 0.001^2])	diag([0.001^2; 0.001^2; 0.001^2])	0.110

4.2 Time of Closest Approach Simulation

In this portion, the update step of the EKF is removed. This means that the only step that remains is the prediction of the state and covariance. This is similar to a situation in which only a couple of measurements are obtained at one time step and the dynamics must be propagated into the future to discover any potential collision with limited knowledge. This would also result in a large and growing covariance without the update step correcting it. This would be done when it is noticed that two objects will come in close contact with each other in the future, and this propagation of the covariance is carried out to determine the associated risk of contact. The pseudo-code for the main code is as follows:

<u>Pseudo-Code:</u>

-Initialize states and covariance variables for EKF
-for k=1: length(time_span)-1
    %Prediction step
    -Predict states forward to the next k+1
    -Predict covariance forward to the next k+1
-end
-Find the index of TCA
-Calculate the P_c using the predicted state and covariance associated with that index

First, the true trajectory under no disturbances was propagated using ode113() to verify that the error of the states is tiny. This will demonstrate the accuracy of the dynamic model to ensure that while the covariance will grow, the predicted state itself does not become unreasonable. The predicted state and covariance used an Δt equal to the orbital period of the ISS (about 5.5535e+03 seconds) divided by 2000 and the simulation was run for 1.5 times the period of the ISS.

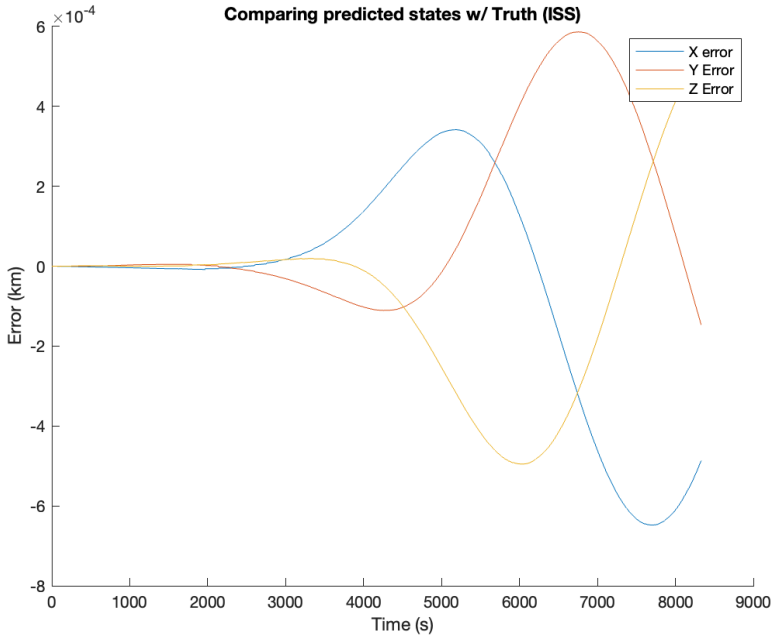


Figure 3: Comparing Error of predicted state vs propagated dynamics

With the error verified to be tiny, the time of closest approach between the predicted states of the two objects can be obtained, and the respective probability of collision calculated. While there exist efficient algorithms to calculate a global minimum distance between the objects, the dynamics in this scenario are periodic and simple. This allows for the global minimum distance to be found via the minimum magnitude of the difference between the positions of the objects and then finding the associated time index. Doing so led to a TCA at 383.1885 seconds into the propagation and a distance of 49.4163 km. The 3-σ covariance was used for the probability calculator, which represents ~99.7 of cases. An upper limit of n=50,000 was used for the infinite series presented in Eq.7.

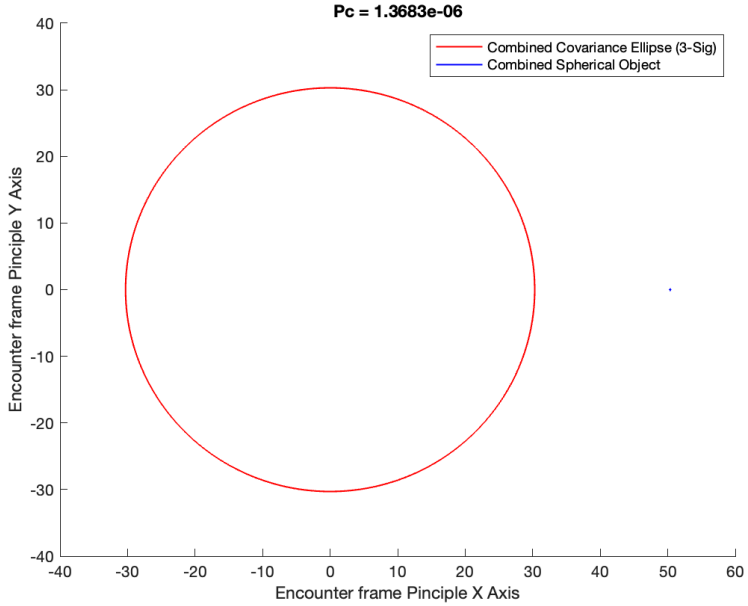


Figure 4: Probability Calculation at TCA

While this probability calculated of 1.3683e-06 is small, it makes sense when it is considered in the context of the two objects in Fig.4 . The combined spherical object is not within the covariance ellipses at all, and yet, the non-zero probability helps quantify just how far it is from being within the covariance ellipse. Additionally, the mitigation threshold typically used is only a couple of orders of magnitude higher. Using this 2-d probability calculation, the acceptable threshold is typically 1e-4 as defined by NASA as being widely acceptable in the industry and historically provides a good balance between safety and mission impact [2]. Thus, the calculated probability is quite reasonable and in this scenario, a corrective maneuver is not necessary to avoid the oncoming object for the ISS.

Using this simulation, a relationship between the probability calculator and the upper limit for the infinite series can be found. This was done at the specified TCA to obtain a lower number to be used for future calculations (Fig 5).

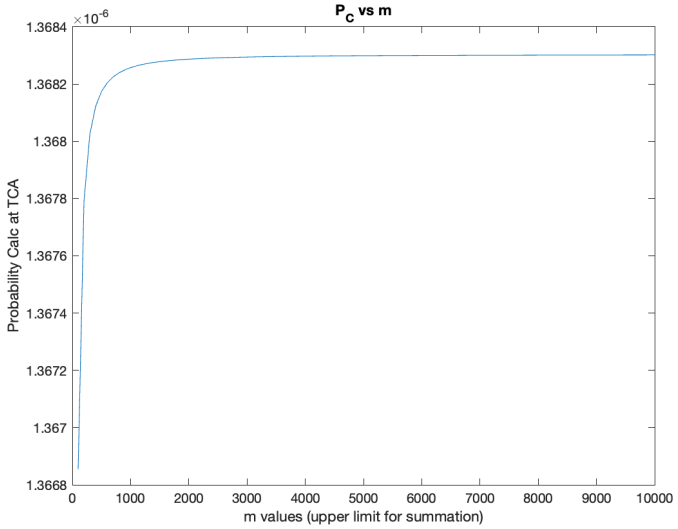


Figure 5: Probability Calculation vs Upper Limit of Summation

From the plot, it can be seen that the probability calculation coverages to a steady state value much sooner than the m=50,000 that was used before. The original paper by Alfano mentions an upper limit of 50 for the power series version, which seems to be a pretty reasonable guess for the actual series given the figure above. However, to ensure the probability calculation fully settles, an upper limit of 2000 was used in the next section.

4.3 Full Orbit Simulation

An interesting application of all this is the ability for onboard calculation of the probability of collision as part of the EKF filter loop. This would allow for the prediction of a collision probability for the current time and continuously update this probability as more information comes in from sensors. The EKF would then fuse this information to provide an updated smaller covariance to be propagated forward and a new probability calculated. This would allow for a much better assessment of the risk of collision as the propagation of the state/covariance from just the initial state (such as section 4.2) doesn't provide the full picture. Having the prediction step run along the EKF filter allows for a more accurate assessment of the risk without having to prematurely make a decision. This allows for the probability calculator to take into account any disturbances that will act on both objects or unexpected changes to either state that would make the initial calculator inaccurate. This delays the time required to make a

decision and rewards the delay with a better-informed depiction of the total risk. The pseudo-code for this section is as follows:

Pseudo Code:

-Initialize ACTUAL states and covariance variables for EKF
-for k=1: length(time_span)-1
    %Prediction step
    -Predict states forward to the next k+1
    -Predict covariance forward to the next k+1

    %Kalman gain
    -Kalman Gain for both objects

    %Update step
    -Update states with Kalman gain for k+1
    -Update covariance with Kalman gain for k+1

    -Initialize a NEW set of states and covariance variables as Update step k+1.
    %Propagate NEW state/covariance until the end of the time span,
    %Note next step k+1 → l-k+2, and current step k→l-k+1
    -for l=k:length(time_span)-1
        %Prediction Step
        -Predict the state of new variables forward to the next l-k+2
        -Predict the covariance of new variables forward to the next l-k+2
    -end

    -Find the index of TCA of NEW state/covariance variables
    -Calculate the P_C using the predicted state and covariance associated with that index
-end

This simulation was run with the same Δt as Section 4.2 but with a decreased final time from 1.5 times the orbital period to 1 orbital period of the ISS. The entire simulation with these parameters took 950.62 seconds.

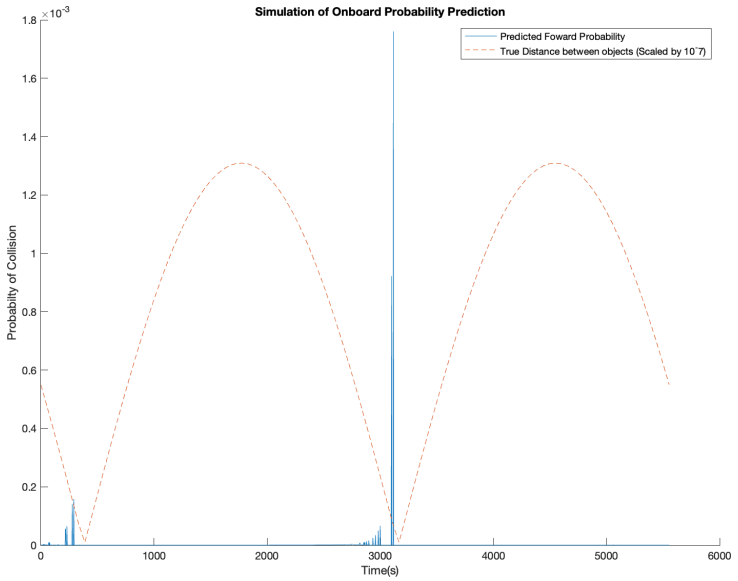


Figure 6: Predicted forward probability w/ scaled down miss distance between objects

Overall, the results are as expected. As the true miss distance between the objects reaches a minimum, the probability of a collision spikes up. Additionally, leading up to the close encounters, the probability also starts to climb before going back down. However, the probability should have peaked higher in the first instant of a minimum in the miss distance rather than the second. This is because as time goes on, the starting covariance being propagated is shrinking as the filter runs, resulting in a lower max probability as the filter runs. For now, this could be propped up to randomness since leading up to the massive peak at ~3100 seconds, the growth in probability was much less than the growth in probability at ~200 seconds. This indicates that at this second close encounter, the Kalman filter estimated the positions of the spacecraft to be closer together than they actually were. To test this, a rng seed of 101 was chosen to run the simulation again with (Fig.6 was run seed set to 1).

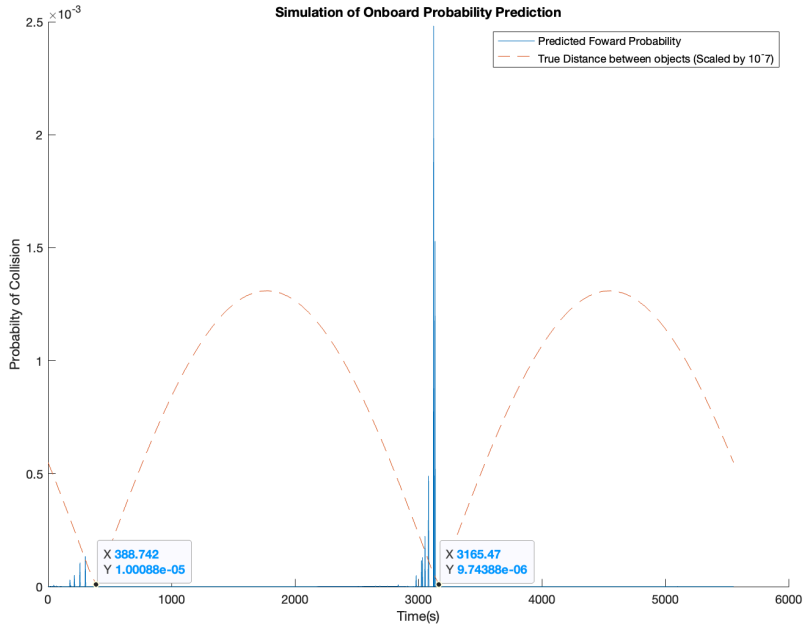


Figure 7: Predicted forward probability w/ scaled down miss distance between objects (seed=101)

From this second run, it is still hard to exactly rule out any other reason for the discrepancy but it's clear that randomness does play a factor. Not only did the max probability over this period increase, but looking at the two minimum peaks, there is also a difference between the magnitudes. Ideally, due to how the initial conditions were picked and with both remaining in a 2-body orbit, the minimum distance between the two spacecraft should happen at the same two places at the same distance. But in this case, the second peak ended up having a smaller minimum distance. This is only amplified by the fact that the values seen in Fig.7 are scaled down by a factor of 10^-7, meaning the difference is much bigger and does potentially explain why the second peak in probability was so much higher.

5. Conclusion

In conclusion, this project was successful in being able to run a collision probability calculator and constantly update it with new information via an EKF. This could readily be implemented onboard or from a mission control center to track the probability of collision of a chosen spacecraft. This could also easily be scaled up to track multiple objects, given enough computational power. In cases like these, an onboard implementation doesn't make the most sense, and this should be run offline. While it could not be confirmed that the probability calculation of Alfano's method was fully recreated in this report, all possible avenue was taken to validate and ensure the content and correctness matched up. From checking the max iteration of the series and determining a minimum limit to checking the magnitude of the probabilities calculated against standard thresholds used in industry, to cross-checking assumptions made across different conventions in defining the encounter frame, the basis of all 2-d probabilistic models for collision. It is unlikely that there exists a big discrepancy,y if any, between the method recreated here and the one described in his original paper.

Regardless, there are a few final notes on this that should be made. Firstly, using Euler's linearization for the dynamics leads to a huge dependence on the time step used. Choosing one that is too big would result in the predicted state skipping over a true global minimum (and thus true TCA) and result in a lower probability than if it were included. This could be remedied by decreasing the time step to an even smaller fraction, which was not explored due to the amount of computational time it would take. Another solution could be to switch over to something like a Sigma Point / Unscented Kalman Filter to do the propagation of the covariance/states. While this filter still suffers from selecting a proper step size, it takes in multiple points before averaging and returning a value, as compared to the Euler linearized EKF, which only uses the current state to predict the next.

Another issue was in the setup of the objects having two points of periodic close contacts with each other especially when considering disturbances and noises altering the true state. With the simple method of finding a TCA described, these small alterations could result in the method skipping over the immediate minima, and going to the next one. If the propagation began at t=0, but the code ends up selecting the second / later minima instead of the immediate one, this would artificially produce a spike in the probability since there would be more time for the covariance to be propagated forward and therefore grow. Then once propagated to that second minimum, there will be a larger uncertainty than usual, thus causing a big spike rather than a gradual increase. Theoretically, the stochastic nature of the state should also impact the probability as the longer propagation time would also result in more error in the predicted state, but because the dynamics of the 2-body problem are pretty accurate, this isn't a likely scenario. Cases where a constant probability calculation algorithm is run sees the secondary object just fly by without coming back periodically, or the period in which it comes back is much longer than the time

span where the states are being propagated. Both cases result in only one true global minima and thus would remedy the issues seen in section 4.3. But if two spacecraft coming into close proximity of each other periodically like the set-up here is not enough of a hazard for the respective operators, a solution to this would be to track the probability of both minimas and track them through each propagation loop, such that there is no doubt between which peak is being used as TCA.

Lastly, it is clear that variability in the measurement and disturbance plays a huge role in the final probability calculated. Therefore, this project would benefit immensely from a future addition of a Monte-Carlo simulation of section 4.3 ran for hundreds if not thousands of loops in order to build a full picture of the probability changes across the trajectories of both spacecraft. This is most likely what is done in actual implementations of this project but was not pursued due to one loop taking ~13 minutes to complete, and thus a large number of trials could not be achieved within the allotted time.

6. Bibliography

[1] [Romain Serra, Denis Arzelier, Mioara Joldes, Jean-Bernard Lasserre, Aude Rondepierre, et al.. , Fast and Accurate Computation of Orbital Collision Probability for Short-Term Encounters. Journal of Guidance, Control, and Dynamics, 2016, 39 (5), pp.1009-1021. ff10.2514/1.G001353ff.ffhal-01132149f
URL:
https://hal.science/hal-01132149/document

[2] NASA, "NASA Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook"
URL:
https://nodis3.gsfc.nasa.gov/OCE_docs/OCE_51.pdf

[3] Salvatore Alfano, "COLLISION AVOIDANCE MANEUVER PLANNING TOOL", 15th AAS/AIAA Astrodynamics Specialist Conference
URL:
https://www.agi.com/getmedia/261d7f49-b286-4cee-a896-4bc9dbe1e0fa/Collision-Avoidance-Maneuver-Planning-Tool.pdf?ext=.pdf)