Conservative Algorithms for Automated Collision Awareness for Multiple Unmanned Aerial Systems

Kevin Ueunten, 2ND LT, USAF
William E. Boeing Department of Aeronautics & Astronautics
University of Washington
Seattle, WA 98195
206-543-7748
kueunten@uw.edu

Christopher Lum
William E. Boeing Department of Aeronautics & Astronautics
University of Washington
Seattle, WA 98195
206-685-1182
lum@uw.edu

Al Creigh
Keisuke Tsujita
William E. Boeing Department of Aeronautics & Astronautics
University of Washington
Seattle, WA 98195
206-543-7748
creigha@uw.edu
kei5328@uw.edu

***Abstract*—As the Federal Aviation Administration (FAA) prepares to integrate Unmanned Aerial Systems (UAS) into the National Airspace System (NAS), developing technologies that mitigate the risk associated with UAS collisions have become a top priority. Despite advances in detect and avoid technologies, the UAS operator remains the primary controller responsible for maintaining inter-vehicle separation and ensuring conflicts do not occur. This paper examines a collision awareness system which increases the operator's situational awareness by spatially and temporally predicting conflicts between the UAS and entities such as other aviation traffic or restricted airspaces. By modeling entities as 3D point masses, the system can be implemented for various, dissimilar UASs. Furthermore, the system supports aircraft engaged in different flight modes such as free flight, following a flight path, and orbit/loiter behavior. Mixed Gaussian distributions model each entity's future position, where the mean is determined by 3D kinematic motion and the covariance is determined by a continuous time error propagation model. Convolving these mixed distribution with another entity or airspace yields mathematically conservative future conflict estimates. Scenarios are presented to demonstrate the algorithm's capabilities.**

TABLE OF CONTENTS

1. INTRODUCTION

With the upcoming integration of UAS in the NAS [1], stakeholders such as the FAA and Department of Defense (DoD) [2] are concerned with their safe integration and operation with the primary concern being human safety [3]. Several earlier studies have investigated tools to estimate the risk of general aviation mid-air collisions with other aircraft [4] as well as other airborne obstacles [5]. Some of these results were adapted to focus on UAS specific mission and provide models and tools for assessing risk of UAS missions to bystanders both in the air [6], [7] and on the ground [8]. The eventual goal is to have full detect and avoid (DAA) systems on UAS to provide automatic collision avoidance [9], [10]. Until this technology is verified, validated, and disseminated, UAS operators are currently responsible for maintaining inter-aircraft spacing and ensuring that conflicts and collisions do not occur [11]. Maintaining situational awareness with respect to current and impending conflicts with both aircraft and restricted airspaces can significantly increase operator workload, thereby reducing the operator's bandwidth to focus on other mission tasks such as search and rescue [12], [13], [14] path planning [15], or maintaining formation [16]. Various groups have looked at human/machine interfaces to increase efficiency and synergy between the operator and the automation [17], [18]. These systems can be augmented with collision awareness systems to further reduce operator workload. Today, manned aircraft uses a traffic collision avoidance system (TCAS) to automatically maintain separation distances and increase the pilot's situational awareness [19]. TCAS is highly regarded by the aviation community and pilots are instructed to follow TCAS warnings to prevent collisions [20]. Currently, a system similar to TCAS has not been specifically implemented for UASs.

The algorithms and systems outlined in this paper aim to reduce operator workload with respect to current and impending conflicts. Unlike TCAS, the proposed system has default settings which the operator can modify to vary the sensitivity levels. Furthermore, the system's predictive threat detection capabilities allow for multiple-threat resolution and restricted airspace warnings. These systems can be integrated into existing UAS ground station software such as the Insitu Common Open Mission Command and Control (ICOMC2) [21] to highlight potential future conflicts with the operator's vehicle and other air traffic or restricted areas.

The collision awareness system has two main components: the forward state estimator and the conflict calculator. An aircraft's position is modeled as a mixed Gaussian distribution, which is determined by the forward state estimator. The forward state estimator supports three different flight modes: free flight, flight path and orbit. After calculating each aircraft's forward state estimates, the conflict calculator integrates the distributions and conservatively overestimates the conflict probability between an aircraft and an airspace or another aircraft. Unlike the forward state estimator, the

978-1-4799-5380-6/15/$31.00

conflict calculator accounts for the aircraft's height, wingspan and necessary separation distances, which provide a conservative conflict prediction tailored for a specific aircraft. After discussing the forward state estimator and conflict calculator algorithms, the paper describes two scenarios to illustrate how the overall collision awareness system operates.

2. FORWARD STATE ESTIMATOR

The forward state estimator (FSE) predicts an aircraft's future position in 3D Cartesian space. The aircraft's position is modeled as a Gaussian random variable. Given that the distribution has infinite tails, the operator must specify a confidence level for computations. In most scenarios illustrated in this paper, a 95% confidence level is used[2]. An aircraft's xy-position is modeled a 2D Gaussian distribution, while the z-position is modeled as a separate, independent 1D Gaussian distribution. Every aircraft is modeled as a 3D point mass, where the distribution's mean is determined by 3D kinematic motion and the covariance is determined from a continuous time error propagation model [22]. Furthermore, the FSE supports three different flight modes: free flight, flight path and orbit. An aircraft engaged in following a flight path or an orbit has known intent, thus the FSE result has reasonable constraints based on the flight path or the orbit. An aircraft with unknown intent is in free flight, which assumes the aircraft continues to fly at the current heading and speed.

Required Inputs

For all flight modes the FSE requires the aircraft's current position, velocity, and aircraft specific errors. The parameter errors provided include the aircraft's GPS error in the xy-plane (σ_{x0} and σ_{y0}) and in the z-direction (σ_{z0}), speed error (σ_s), climb angle error (σ_ϕ) and heading angle error (σ_θ). Furthermore the error provided is at the user defined confidence level (95% in subsequent figures). The standard deviation for each parameter variable is calculated using a standard normal distribution.

Further inputs are required for aircraft with known intent. For the flight path mode, the FSE requires a list of waypoints and if the flight path is closed or open. Orbits are modeled as constant altitude circles and thus require the orbit's center, radius and the orbit direction.

3. FORWARD STATE ESTIMATOR FREE FLIGHT

The FSE operating in free flight mode is the default flight mode for all aircraft with unknown intent. Figures 1 and 2 illustrate FSE free flight results, which are visualized as beige cylindrical ellipses. These figures illustrate an aircraft in free flight mode that is climbing at a constant speed. As expected, the aircraft's FSE results increase in both size and altitude as the prediction time increases (thereby illustrating the growing uncertainty in the aircraft's position).

Mean Calculation

An aircraft engaged in free flight continues to fly at its current heading and speed. The speed (s) is the magnitude of the current velocity and the climb angle (ϕ) and θ are calculated based on the current velocity's components.

[2]The confidence level can be operator specified. A larger confidence level yields a more conservative calculation.

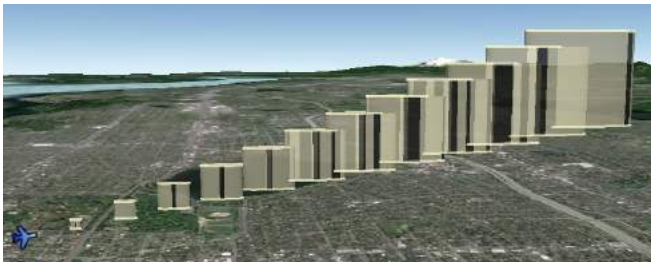

Figure 1. This shows an FSE free flight results from the side. Note that the area of space required to capture the same confidence level grows as the prediction horizon increases.

Figure 2. This shows an FSE free flight results from the top. Notice that errors in the initial velocity estimate will lead to larger regions in the body x-axis whereas errors in the initial flight path heading angle will yield larger propagation in the body y-axis.

$$\begin{array}{l} \phi = \text{atan2}\left(Vel_z, Vel_{xy}\right) \\ \theta = \text{atan2}\left(Vel_y, Vel_x\right) \end{array} \tag{1}$$

Applying these three parameters, an aircraft's mean position at time t can be estimated as

$$\begin{bmatrix} x\left(t\right) \\ y\left(t\right) \\ z\left(t\right) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + st \begin{bmatrix} \cos\left(\phi\right)\cos\left(\theta\right) \\ \cos\left(\phi\right)\sin\left(\theta\right) \\ sin\left(\phi\right) \end{bmatrix} \tag{2}$$

where, $[x_0,\, y_0,\, z_0]^{\mathrm{T}}$ is the initial position and t is the prediction time.

Covariance Calculation

The covariance is calculated using the second order error propagation technique applied to the aircraft's motion in the body coordinate system. Figure 3 defines the body coordinate system and Eq.3 describes the aircraft's motion in the body frame.

$$\begin{bmatrix} x_{body}\left(t\right) \\ y_{body}\left(t\right) \\ z_{body}\left(t\right) \end{bmatrix} = \begin{bmatrix} x_{body}\left(0\right) \\ y_{body}\left(0\right) \\ z_{body}\left(0\right) \end{bmatrix} + st \begin{bmatrix} \cos\left(\theta_{body}\right) \\ \sin\left(\theta_{body}\right) \\ \cos\left(\phi_{body}\right) \end{bmatrix} \tag{3}$$

To account for the worst case scenario the s in Eq.2 is used in Eq.3. For example let the velocity be $[2\frac{\mathrm{m}}{\mathrm{sec}},\ 2\frac{\mathrm{m}}{\mathrm{sec}},\ 1\frac{\mathrm{m}}{\mathrm{sec}}]^{\mathrm{T}}$ thus the speed is $3\frac{\mathrm{m}}{\mathrm{sec}}$. Therefore the speed in the body's x-axis direction is actually $2\frac{\mathrm{m}}{\mathrm{sec}}$ but $3\frac{\mathrm{m}}{\mathrm{sec}}$ will be used. Since the actual speed in each body axis is less than the aircraft's speed, the body motion equations will always capture the worst case scenarios and yield conservative results. Also since the body frame is aligned with the aircraft's motion, both θ_{body} and ϕ_{body} are 0°.

Eq.4 is the general form for the second order error propagation method [22]. This method incorporates the covariance

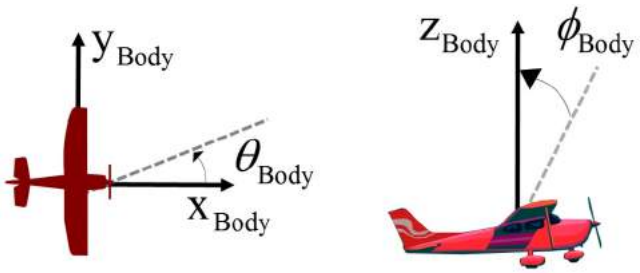

Figure 3. This is the body coordinate system where the $\mathrm{x_{Body}}$-axis starts from the aircraft's center to the aircraft's nose and is aligned with θ and the $\mathrm{y_{Body}}$-axis begins at the aircraft's center and ends at the left wingtip. Furthermore $\mathrm{z_{Body}}$-axis is the cross product of the $\mathrm{x_{Body}}$-axis and $\mathrm{y_{Body}}$-axis. θ_{Body} is the angular deviation from the $\mathrm{x_{Body}}$-axis. Similarly the ϕ_{Body} is the angular deviation from the $\mathrm{z_{Body}}$-axis.

between each parameter and includes the partial derivatives, which result in an accurate, yet conservative standard deviation.

$$\sigma_F{}^2 = \bar{g}_0^T \Sigma_x \bar{g}_0 + \frac{1}{2}\mathrm{trace}\left(\mathrm{H_0}\Sigma_\mathrm{x}\mathrm{H_0}\Sigma_\mathrm{x}\right) \tag{4}$$

Applying second order error propagation techniques to the aircraft's motion in the body frame yields

$$\begin{gathered}\bar{g}_0 = \begin{bmatrix} 1 \\ tsin\left(\theta_{body}\right) \\ st\cos\left(\theta_{body}\right) \end{bmatrix}, \Sigma_x = \begin{bmatrix} \sigma_{x_0}^2 & 0 & 0 \\ 0 & \sigma_s^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}, \\ H_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & t\cos\left(\theta_{body}\right) \\ 0 & t\cos\left(\theta_{body}\right) & -st\sin\left(\theta_{body}\right) \end{bmatrix}\end{gathered} \tag{5}$$

The continuous time error propagation equations are

$$\begin{aligned}\sigma_{x,Body} &= \sqrt{\frac{\sigma_\theta^4 s^2 t^2}{2} + \sigma_{x_0}^2 + \sigma_s^2 t^2} \\ \sigma_{y,Body} &= \sqrt{\frac{\sigma_\theta^2 \sigma_s^2 t^2}{2} + \sigma_{y_0}^2 + s^2\sigma_\theta^2 t^2} \\ \sigma_{z,Body} &= \sqrt{\frac{\sigma_\phi^4 s^2 t^2}{2} + \sigma_{z_0}^2 + \sigma_s^2 t^2}\end{aligned} \tag{6}$$

Since no rotation is required between the body and Cartesian system the variance in the z-direction is the same.

$$\sigma_z^2\left(t\right) = \sigma_{z,Body}^2\left(t\right) \tag{7}$$

Assuming $\sigma_{x,Body}$ and $\sigma_{y,Body}$ are independent the covariance is

$$Cov_{Body}\left(t\right) = \begin{bmatrix} \sigma_{x,Body}^2\left(t\right) & 0 \\ 0 & \sigma_{y,Body}^2\left(t\right) \end{bmatrix} \tag{8}$$

A direction cosine matrix (DCM) with rotation angle θ transforms the covariance from the body frame to the Cartesian system.

Figure 4. This shows two FSE flight path results: a parallel track (left) and expanding square flight path (right).

$$Cov_{XY} = DCM_{Body/XY}^{-1} Cov_{Body} DCM_{Body/XY} \tag{9}$$

4. Forward State Estimator Flight Path

The FSE flight path conservatively predicts an aircraft's future position for an aircraft engaged in flight path following mode. A flight path is defined by a list of waypoints in 3D Cartesian space and the line between each two consecutive waypoints generates the flight path leg. The FSE flight path assumes an aircraft flies at a constant speed on the flight route and can instantaneously turn at waypoints. Figure 4 illustrates two different flight paths and their associated FSE flight path results.

Mean Calculation

A 3D kinematic point mass model describes the aircraft's motion. However since the aircraft follows a flight path, θ and ϕ changes based on the flight path leg. The waypoint to which the aircraft is heading at time t determines θ and ϕ. $t_{waypoint_K}$ is the time the aircraft takes to reach waypoint K.

$$t_{waypoint_K} = \frac{dist_{waypoint}}{s} \tag{10}$$

$dist_{waypoint}$ is the distance between $pt1$ ($waypoint_K$) and $pt2$ ($waypoint_{K-1}$). When $K = 1$, $waypoint_{K-1}$ is the aircraft's current position.

t_{sum} determines the waypoint the aircraft is headed towards at time t.

$$t_{sum} = \sum_{K=1}^{N} t_{waypoint_K} \tag{11}$$

where N is the first waypoint such that $t_{sum} > t$.

Thus, $waypoint_N$ is the waypoint the aircraft heads to at time t. Using N,

$$\theta = \text{atan2}\left(pt2_y - pt1_y,\ pt2_x - pt1_x\right) \tag{12}$$

$$\phi = \text{atan2}\left(pt2_z - pt1_z,\ dist_{xy}\right) \tag{13}$$

where $dist_{xy}$ is the xy-distance between $pt2$ and $pt1$.

$t_{current}$ is

$$t_{current} = t - t_{sum-1} \tag{14}$$

where

$$t_{sum-1} = \sum_{K=1}^{N-1} t_{waypoint\,K} \tag{15}$$

Substituting θ, ϕ and $t_{current}$ into Eq.2 calculates the predicted mean position at t.

Covariance Calculation

The aircraft's covariance is calculated in the body frame and transformed into the Cartesian frame. In the body frame the x-axis is aligned with the flight path leg and the y-axis is perpendicular to the flight path leg. It is reasonable to expect that the aircraft will not deviate in the body y-axis. This expectation stems from the fact we assume the aircraft's inner loop controller will forces the aircraft to follow the flight path. Furthermore this inner loop controller is expected to maintain the correct altitude thus the error in the body frame's z-axis is expected to remain constant. Using these assumptions the body frame's error in the y-direction and the z-direction is

$$\begin{aligned} \sigma_{y\ Body} &= \sigma_{y_0} \\ \sigma_{z\ Body} &= \sigma_{z_0} \end{aligned} \tag{16}$$

σ_x is calculated using first order error propagation techniques [22] because $\sigma_\theta = 0$.

$$\sigma_{x,Body} = \sqrt{\sigma_{x_0}^2 + (\sigma_s t)^2} \tag{17}$$

Once $\sigma_{x,Body}$, $\sigma_{y,Body}$, and $\sigma_{z,Body}$ are calculated, the same process used by the FSE free flight transforms the covariance from the body frame to the Cartesian frame.

Open and Closed Flight Paths

The algorithm handles the open and closed flight paths. If the flight path is open, the aircraft remains at the last waypoint and $\sigma_{x,Body}$ stops growing once the aircraft reaches the last waypoint. If the flight path is closed, the aircraft proceeds to the first waypoint after reaching the last waypoint and $\sigma_{x,Body}$ continues to grow. Figure 5 illustrates an open and closed flight path.

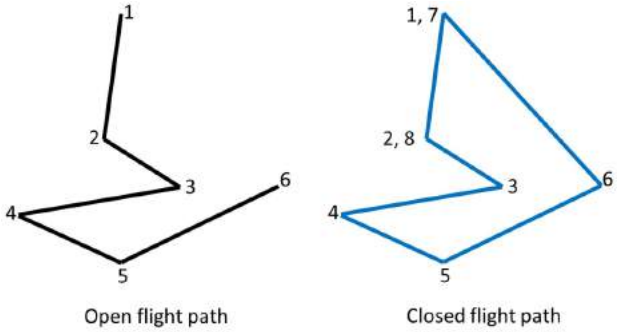

Figure 5. This shows an open flight path and a closed flight path including ascending waypoint numbers.

5. FORWARD STATE ESTIMATOR ORBIT

The FSE orbit conservatively predicts an aircraft's future position for an aircraft engaged in an orbit. An aircraft in orbit mode flies in a circular pattern exclusively clockwise or counterclockwise at a constant altitude and speed. Figures 6 and 7 illustrate an FSE orbit result. In these figures the aircraft is on the orbit in the xy-plane however is at a higher altitude than the orbit altitude. Thus the aircraft will descend while following the orbit in the xy-plane until the orbit altitude is reached.

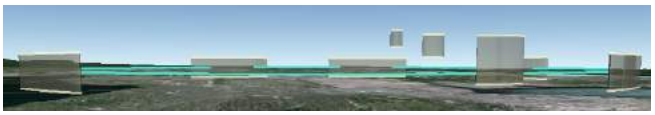

Figure 6. This is the side view of FSE orbit result. Notice the altitude change in the estimates as the aircraft adjust altitude to meet the orbit.

Figure 7. This is the top view of FSE orbit result. Notice how the covariance increases in the body x-axis but remains bounded in the body y-axis.

Mean Calculation

The mean calculation is separated into two cases: aircraft begins on the orbit or does not begin on the orbit. Let the orbit center be $[x_{orbit},\, y_{orbit},\, z_{orbit}]^{\mathrm{T}}$. First $dist_{xy}$ and alt_{dif} are calculated to determine if the aircraft begins on the orbit or not.

$$dist_{xy} = dist_{xy,Orbit} - orbit_{radius} \tag{18}$$

where $dist_{xy,Orbit}$ is the distance in the xy-plane between the orbit center and the aircraft's current position.

$$alt_{dif} = z_{orbit} - z_0 \tag{19}$$

If both $dist_{xy} = 0$ and $alt_{dif} = 0$, then the aircraft is on the orbit. If not, then the aircraft is not on the orbit (this case is discussed later in this section).

θ_0 is defined the same way polar coordinate angles are.

$$\theta_0 = \operatorname{atan2}\left(\left(y_0 - y_{orbit}\right), \left(x_0 - x_{orbit}\right)\right) \tag{20}$$

Aircraft is on the orbit—In this case, θ_{trav} is the angle the aircraft traveled on the orbit.

$$\theta_{trav} = \frac{st}{orbit_{radius}} \tag{21}$$

The aircraft's predicted position depends on the orbit direction traveled.

$$\begin{array}{l} \text{Counterclockwise} : \theta(t) = \theta_0 + \theta_{trav} \\ \text{Clockwise} : \theta(t) = \theta_0 - \theta_{trav} \end{array} \tag{22}$$

Eq.23 transforms the angular position to the mean position.

$$\begin{array}{l} x\left(t\right) = x_{orbit} + \cos\left(\theta\left(t\right)\right) orbit_{radius} \\ y\left(t\right) = y_{orbit} + \sin\left(\theta\left(t\right)\right) orbit_{radius} \\ z\left(t\right) = z_{orbit} \end{array} \tag{23}$$

Aircraft is not on the orbit—There are three different situations when the aircraft is not on the orbit:

1. On the orbit in the xy-plane but at a different altitude.
2. Outside the orbit.
3. Inside the orbit.

An aircraft is inside an orbit when $dist_{xy} < 0$ and outside the orbit when $dist_{xy} > 0$.

This algorithm assumes an aircraft not on the orbit proceeds directly to the orbit and enters the orbit following a radial path as illustrated in Figure 8. Another assumption is an aircraft not at the orbit altitude flies at a climb angle (ϕ) to reach the orbit altitude, however ϕ cannot exceed a max climb angle[3] for an aircraft not at the orbit altitude. Although the system uses these assumptions, there are other methods to model an aircraft's motion, when the aircraft is not on the orbit.

[3]For this paper the max angle is $10°$.

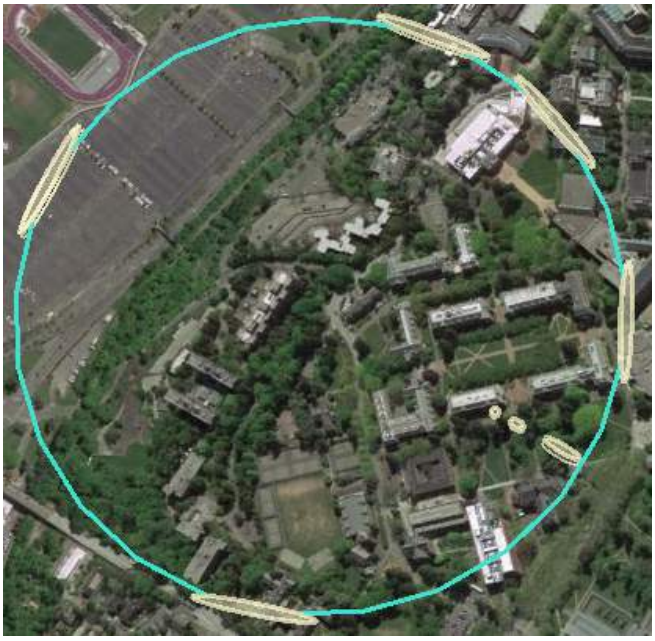

Figure 8. Aircraft is initially inside the orbit and travels radially outward until the orbit is reached. Then the FSE results tracks the aircraft's position on the orbit.

Since there is a specified orbit altitude, ϕ has to be carefully defined. The case when $dist_{xy}$ is discussed in the next section.

$$\phi_0 = \left|\operatorname{atan}\left(\frac{alt_{dif}}{dist_{xy}}\right)\right| \Rightarrow \begin{cases} \text{if } alt_{dif} < 0,\ \phi_{initial} = -\phi_0 \\ \text{if } alt_{dif} > 0,\ \phi_{initial} = \phi_0 \end{cases} \tag{24}$$

Applying the max climb angle assumption, if $|\phi_{initial}| \leq 10°$, then $\phi = \phi_{initial}$. The other case is if $|\phi_{initial}| > 10°$, then $\phi = \pm 10°$ the sign has the same sign as $\phi_{initial}$.

t_{alt} is the time the aircraft reaches the correct orbit altitude and t_{orb} is the time the aircraft reaches the orbit in the xy-plane.

$$t_{alt} = \frac{alt_{dif}}{s\sin(\phi)} \tag{25}$$

$$t_{orb} = \frac{|dist_{xy}|}{s\cos(\phi)} \tag{26}$$

Situation 1: Aircraft on Orbit in xy-Projection but Different Altitude

There are two different cases for this situation: an aircraft at a higher or lower altitude.

When the aircraft is above the altitude, the aircraft follows the orbit in the xy-plane and $\phi = -10°$. For the other case the aircraft performs the same action except $\phi = 10°$. Also there are two distinct time periods, $t < t_{alt}$ and $t > t_{alt}$. When $t < t_{alt}$, the z-position is

$$z(t) = z_0 + st\sin(\phi) \tag{27}$$

To calculate the xy-position a new angle is introduced, θ_{trans}, which is the angle the aircraft travels in the orbit while reaching the orbit altitude.

$$\theta_{trans} = \frac{s\left|\cos(\phi)\right| t}{orbit_{radius}} \tag{28}$$

θ is calculated by substituting θ_{trans} for θ_{trav} in Eq.22. $x(t)$ and $y(t)$ is calculated using Eq.23.

When $t > t_{alt}$, $\theta_{transNotAlt}$, the angle the aircraft traveled from the initial time to t_{alt}, is calculated by substituting t_{alt} for t in Eq.28. Then $\theta_{transOnOrb}$, the angle the aircraft traveled at $t - t_{alt}$ is calculated using $t - t_{alt}$ in Eq.21. $\theta(t)$ is calculated using Eq.22 and Eq.29. Then, the mean position is calculated using Eq.23 and $\theta(t)$.

$$\theta_{trav} = \theta_{transNotAlt} + \theta_{transOnOrb} \tag{29}$$

Situation 2: Aircraft is Outside the Orbit

Although there are three different cases for this situation: aircraft is at a higher altitude, lower altitude or at the orbit altitude, the logic for each case is the same. For this situation there are three distinct time periods: $t < t_{orb}$, $t < t_{alt}$ and $t > t_{alt}$. Remember $t_{orb} \leq t_{alt}$.

When $t < t_{orb}$, the aircraft heads directly to the orbit and enters the orbit radially the aircraft follows the angle ψ.

$$\psi = \theta_0 + 180^\circ \tag{30}$$

Thus, the mean is calculated using Eq.2 and substituting ψ for θ.

When $t < t_{alt}$, the aircraft is on the orbit in the xy-plane, which is similar to situation 1. θ_{trans} is calculated by replacing t with $t - t_{orb}$ into Eq.28. Then the mean is calculated using Eq.23 and Eq.27.

When $t > t_{alt}$, the same algorithm used in situation 1 during the same time period is applied for this situation, except in Eq.28 $t_{alt} - t_{orb}$ is substituted for $t - t_{alt}$.

Situation 3: Aircraft is Inside the Orbit

This situation follows the same logic as situation 2 except substitute ψ_{inside} for ψ.

$$\psi_{inside} = \theta_{initial} \tag{31}$$

Covariance Calculation

The FSE orbit uses the same error propagation technique as the FSE free flight, however since the aircraft's intent is known reasonable limits are applied to each body frame direction. The body frame directions are altitude (body frame's +z-axis), orbit's radial direction (body frame's +y-axis), and orbit's tangential direction (body frame's +x-axis). The xy-covariance body frame limits are applied once the aircraft is on the orbit in the xy-plane and the altitude limits are applied once the aircraft reaches the orbit altitude.

A reasonable assumption[4] is the error in the tangential direction (σ_{xError}) cannot exceed the orbit's diameter. Scenarios in this paper use $\sigma_{yError} = 5\text{m}$ and $\sigma_{yError} = 20\text{m}$ but these may be tailored to the specific aircraft system of interest.

Eq.32 to Eq.34 calculate the time it takes to reach the max error in each direction. For these calculations to obtain real solutions $\sigma_{xError} > \sigma_{x0}$, $\sigma_{yError} > \sigma_{y0}$, and $\sigma_{zError} > \sigma_{z0}$. These conditions are reasonable because if the GPS error is larger than the max error, then an aircraft cannot be guaranteed to be within a max error.

$$t_{xError} = \frac{\sigma_{xError}^2 - \sigma_{x_0}^2}{\frac{\sigma_\theta^4 s^2}{2} + \sigma_s^2} \tag{32}$$

$$t_{yError} = \frac{\sigma_{yError}^2 - \sigma_{y_0}^2}{\frac{\sigma_\theta^2 \sigma_s^2}{2} + s^2\sigma_\theta^2} \tag{33}$$

$$t_{zError} = \frac{\sigma_{zError}^2 - \sigma_{z_0}^2}{\frac{\sigma_\phi^4 s^2}{2} + \sigma_s^2} \tag{34}$$

The max error times determine when the max error in each direction is applied. Furthermore, the max error times in the tangential and radial directions are applied when the aircraft is on the orbit in the xy-plane and the max error time in the altitude direction is applied when the aircraft is at the orbit altitude. There are cases when the time it takes to reach the correct altitude and, or orbit is greater than the max error time. For these cases, the error grows normally until the orbit is reached. Once the aircraft reaches the orbit then the max error times are applied.

Also the FSE orbit's DCM rotation angle depends on the orbit direction and whether an aircraft is outside or inside the orbit. If the aircraft is not on the orbit in the xy-plane, ψ and ψ_{inside} is used appropriately. The DCM's rotation angle on the orbit in the xy-plane is defined as

$$\begin{array}{l} \text{Counterclockwise}: \ \theta_{rot} = \theta + \frac{\pi}{2} \\ \text{Clockwise}: \ \theta_{rot} = \theta - \frac{\pi}{2} \end{array} \tag{35}$$

6. CONFLICT CALCULATOR

The conflict calculator requires an FSE result (parameterized as a 3D distribution), aircraft specific information, and airspace properties to determine the conflict probability between an operator-specified aircraft (the perspective entity), and another aircraft or an airspace. Jointly integrating the Gaussian distribution yields the probability of conflict between the perspective entity and another entity. Conservative results are achieved by over-bounding or making overestimates.

The following subsections describe how a 2D and 1D probability is calculated as well as rotating an FSE result's 2D covariance matrix back to the 2D body frame. These techniques

[4] These max error values can be operator specified.

are fundamental for both the aircraft conflict calculator and airspace conflict calculator.

Gaussian Distributions

A 2D Gaussian distribution (Eq.36) represents the FSE result's covariance ($\Sigma(t)$) and mean ($\mu(t)$) in the xy-plane.

$$p_{xy}(t) = \left[(2\pi)^2 \det(\Sigma(t))\right]^{-\frac{1}{2}} e^{-\frac{1}{2}\left(\begin{bmatrix} x \\ y \end{bmatrix}-\mu(t)\right)^T \Sigma(t)^{-1}\left(\begin{bmatrix} x \\ y \end{bmatrix}-\mu(t)\right)} \tag{36}$$

where,

$$\mu(t) = \begin{bmatrix} \bar{x}(t) \\ \bar{y}(t) \end{bmatrix}$$
$$\Sigma(t) = \begin{bmatrix} \sigma_x^2(t) & \operatorname{cov}(x(t), y(t)) \\ \operatorname{cov}(x(t), y(t)) & \sigma_y^2(t) \end{bmatrix} \tag{37}$$

A 1D Gaussian distribution (Eq.38) represents the FSE result in the z-direction.

$$p_z(t) = \frac{e^{\frac{-(z-\bar{z}(t))^2}{2\sigma_z^2(t)}}}{\sqrt{2\pi\sigma_z^2(t)}} \tag{38}$$

The probability in the xy-plane (Eq.40) is calculated by integrating a 2D probability distribution and a diagonal covariance matrix of the form

$$\Sigma = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \tag{39}$$

Since the FSE's body frame's covariance has this diagonal covariance form, this technique is valid. Further discussion on rotating an FSE covariance back to the body frame is described in the next section. Using this diagonal covariance matrix, a closed form expression for the double integral over a box region can be obtained as shown by Eq.40.

$$\begin{aligned} P_{xy}(t) &= \int_{y_0}^{y_f}\int_{x_0}^{x_f} \left[2\pi \det(\Sigma(t))\right]^{-\frac{1}{2}} \\ &e^{-\frac{1}{2}\left(\begin{bmatrix} x \\ y \end{bmatrix}-\mu(t)\right)^T \Sigma(t)^{-1}\left(\begin{bmatrix} x \\ y \end{bmatrix}-\mu(t)\right)} dxdy \\ &= \tfrac{1}{4}\left(\operatorname{Erf}\left(\tfrac{x_f-\bar{x}(t)}{\sigma_x(t)\sqrt{2}}\right) - \operatorname{Erf}\left(\tfrac{x_0-\bar{x}(t)}{\sigma_x(t)\sqrt{2}}\right)\right) \\ &\left(\operatorname{Erf}\left(\tfrac{y_f-\bar{y}(t)}{\sigma_y(t)\sqrt{2}}\right) - \operatorname{Erf}\left(\tfrac{y_0-\bar{y}(t)}{\sigma_y(t)\sqrt{2}}\right)\right) \end{aligned} \tag{40}$$

Next taking the single integral in the altitude direction yields the probability being in the following altitude range as described in Eq.41.

$$\begin{aligned} P_z(t) &= \int_{z_0}^{z_f} \frac{e^{\frac{-(z-\bar{z}(t))^2}{2\sigma_z^2(t)}}}{\sqrt{2\pi\sigma_z^2(t)}} dz \\ &= \tfrac{1}{2}\left(\operatorname{Erf}\left(\tfrac{z_f-\bar{z}(t)}{\sigma_z(t)\sqrt{2}}\right) - \operatorname{Erf}\left(\tfrac{z_0-\bar{z}(t)}{\sigma_z(t)\sqrt{2}}\right)\right) \end{aligned} \tag{41}$$

Rotating the Covariance Matrix in the xy-Plane

Rotating any 2D covariance matrix appropriately, yields a zero off diagonal covariance matrix. To obtain this covariance matrix, first θ, the angle from the 2D Gaussian's semimajor axis to the body frame's +x-axis, is calculated. Using a DCM with rotation angle θ yields a zero off diagonal covariance matrix (Cov_{rot}).

7. AIRCRAFT CONFLICT CALCULATOR

The aircraft conflict calculator is responsible for determining the probability of an aircraft violating another aircraft's airspace. This module allows the system to conservatively predict if two aircraft are in conflict and determine both when and where a conflict will occur. These calculations are done from the perspective of an aircraft that the operator is interested in. This aircraft is deemed the 'perspective entity' and is typically the aircraft that the operator is controlling but any vehicle can assume the role of the perspective entity. The aircraft conflict calculator models aircraft as a 3D point mass and incorporates both the aircraft geometry (such as wingspan and height) and required horizontal and vertical separation distances. An aircraft's airspace is modeled as a cylinder whose radius is a function of the the non-perspective entity's wingspan and the desired level of confidence of calculation accuracy, and required horizontal separation distance (see Eq.53). The height of the cylinder is computed in a similar fashion using parameters related to the vertical separation. For this discussion aircraft A is the perspective entity and aircraft B is the non-perspective entity aircraft. The probability in the planar and altitude direction is assumed independent so the total conflict probability is the product of these two probabilities.

Aircraft Conflict Probability in xy-Plane

To formulate the problem from viewpoint of aircraft A, the following parameters will be used

1. b_A is aircraft A's wingspan, which is the the actual horizontal distance taken up by aircraft A (not the half wingspan).
2. r_A is aircraft A's horizontal separation distance.
3. μ_A is aircraft A's mean position.
4. Σ_A is aircraft A's 2D distribution's covariance.

Before proceeding further, the local world frame F_W is defined with an x_1-axis aligned with the local north direction, x_2-axis aligned with the local east direction, and the origin is the world frame's center. This defines a local Cartesian frame which is similar to the vehicle carried north-east-down frame, except the origin is at an arbitrary "base" location.

Formulating a description of aircraft A's horizontal footprint is important because the aircraft's planar position is stochastic. Thus this horizontal footprint needs to be conservative to account for regions of high likelihood. Note that it is unreasonable to attempt to account for 100% of the possibilities of aircraft A's footprint because under a Gaussian distribution, this requires the range of $x_1 \in [-\infty, \infty]$ and $x_2 \in [-\infty, \infty]$. Instead, consider that aircraft A's footprint can be a circle of

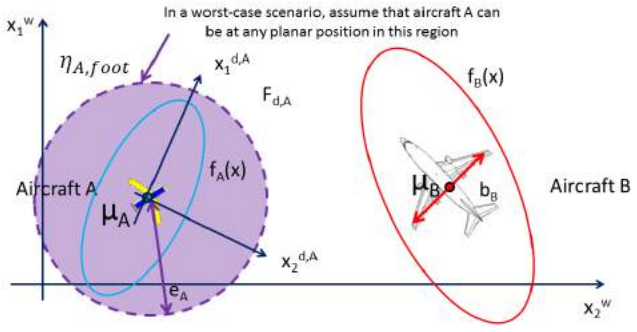

Figure 9. The blue and red ellipses represent the distributions $f_A(x)$ and $f_B(x)$, respectively. Aircraft A's (perspective entity) worst-case horizontal footprint is shown as $\eta_{A,foot}$ (purple circle) with a radius of e_A.

radius e_A and centered around μ_A. Mathematically, this is given by

$$\eta_{A,foot} = \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} | (x_1 - \mu_{A,1})^2 + (x_2 - \mu_{A,2})^2 \leq {e_A}^2 \right\} \tag{42}$$

where $\eta_{A,foot}$ is aircraft A's worst-case horizontal footprint range and e_A is the radius of the worst-case horizontal footprint range. Both variables are illustrated in Figure 9. e_A needs to be chosen large enough to capture the desired confidence level of aircraft A's horizontal positions. A conservative computation of e_A will now be derived.

Integration of f_A over $\eta_{A,foot}$ is used to determine if e_A is large enough to contain the specified percentages of cases. To do this a new frame based on aircraft A's probability distribution function ($F_{d,A}$) is created. This frame has the x_1-axis aligned with distribution f_A's semi-major axis, the x_2-axis aligned with distribution f_A's semi-minor axis, and the origin is at distribution f_A's mean (μ_A). Geometrically, distribution f_A can be expressed in $F_{d,A}$ as

$$f_A^{d,A}(x) = N\left(0, \Sigma_A^{d,A}\right) \tag{43}$$

where N is a 2D Gaussian distribution and

$$\Sigma_A^{d,A} = \begin{bmatrix} \lambda_{A,\max} & 0 \\ 0 & \lambda_{A,\min} \end{bmatrix} \tag{44}$$

In the previous expression, $\lambda_{A,min}$ and $\lambda_{A,max}$ denote the minimum and maximum eigenvalues of the 2D covariance matrix, respectively.

Now integrating $f_A^{d,A}(x)$ over $\eta_{A,foot}$ obtains a cumulative distribution function as a function of the radius, e_A. This integration is

$$F(e_A) = \int\int_{\eta_{A,foot}} f_A^{d,A}(x)\, dA \tag{45}$$

Due to the fact that it is difficult to obtain a closed form solution of a general 2D Gaussian distribution over a circular domain of integration, we approximate the distribution as

$$\tilde{f}_A^{d,A}(x) = N\left(0, \tilde{\Sigma}_A^{d,A}\right) \tag{46}$$

where

$$\tilde{\Sigma}_A^{d,A} = \begin{bmatrix} \lambda_{A,\max} & 0 \\ 0 & \lambda_{A,\max} \end{bmatrix} \tag{47}$$

This distribution, $\tilde{f}_A^{d,A}(x)$, is the same as $f_A^{d,A}(x)$ except that it is symmetric about the origin (because $\lambda_{A,max}$ is on both diagonal elements). In a sense, this is "more spread out" than $f_A^{d,A}(x)$. We can think of $\tilde{f}_A^{d,A}(x)$ as more conservative than the original distribution $f_A^{d,A}(x)$.

We can integrate this (in polar coordinates) over $\eta_{A,foot}$

$$\tilde{F}(e_A) = \int_0^{e_A}\int_0^{2\pi} \tilde{f}_A^{d,A}(x)\, r dr d\theta \tag{48}$$

where $x = \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta) \end{bmatrix}$

to obtain a closed form and conservative expression for the cumulative distribution function

$$\tilde{F}(e_A) = 1 - e^{\frac{-e_A^2}{2\lambda_{A,max}}} \tag{49}$$

Next e_A is solved for using Eq.49.

$$e_A = \sqrt{-2\lambda_{A,\max} \ln\left(1 - \tilde{F}\right)} \tag{50}$$

where $\tilde{F} \in [0,1]$ is the desired confidence level. In practice the bounds of $\tilde{F}$ should be exclusively $(0,1)$ because $\tilde{F} = 0$ implies $e_A = 0$ and $\tilde{F} = 1$ implies $e_A = \infty$, which are both somewhat unreasonable.

Now that we have a description of aircraft A's horizontal footprint, we need to describe the airspace that it claims as its own. We calculate this as

$$\eta_{A,air,h} = \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} | (x_1 - \mu_{A,1})^2 + (x_2 - \mu_{A,2})^2 \leq (e_A + r_A)^2 \right\} \tag{51}$$

$\eta_{A,air,h}$ is aircraft A's worst-case horizontal airspace range and describes the planar positions that if any object comes into this region, we call this an airspace violation.

From geometry, we see that if aircraft B's planar position is anywhere in the range

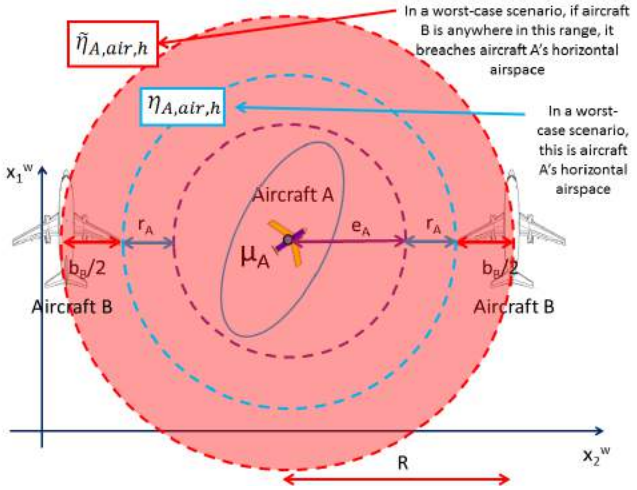

Figure 10. The red circle shows $\tilde{\eta}_{A,air,h}$. If any portion of aircraft B enters this region, we consider it a violation of aircraft A's (the perspective entity) horizontal airspace.

$$\tilde{\eta}_{A,air,h} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in (x_1 - \mu_{A,1})^2 + (x_2 - \mu_{A,2})^2 \leq R^2 \tag{52}$$

where R is given by

$$R = e_A + r_A + \frac{b_B}{2} \tag{53}$$

then aircraft B has the potential to breach aircraft A's horizontal airspace as shown in Figure 10.

Now the probability that aircraft B will breach aircraft A's horizontal airspace can be calculated. $\tilde{\eta}_{A,air,h}$ gives a planar region of space where if aircraft B is anywhere in this region, we consider it a breach of aircraft A's horizontal airspace. Therefore, the probability of this occurring is simply the integral of aircraft B's distribution (f_B) over this domain.

$$p\left(X_B \in \tilde{\eta}_{A,air,h}\right) = \int\int_{\tilde{\eta}_{A,air,h}} f_B\left(x\right) dA \tag{54}$$

As we saw earlier, calculations become simpler if a frame aligned with the distribution f_B is used. f_B is defined as the frame of aircraft B's probability distribution function where the frame's origin is at f_B's mean, x_1-axis is aligned with f_B's semi-major axis and x_2 is aligned with f_B's semi-minor axis.

The vector from aircraft B to aircraft A is

$$r_{A/B} = \mu_A - \mu_B \tag{55}$$

$r_{A/B}$ is most likely expressed in the world frame, F_W. However, this can easily be expressed in $F_{d,B}$ (aircraft B's distribution frame) using

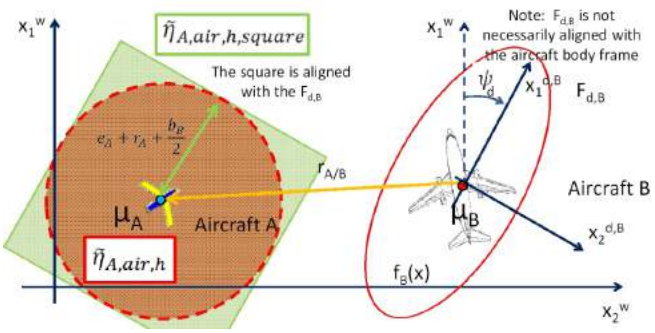

Figure 11. To conservatively find the probability of aircraft B violating the aircraft A's airspace in the xy-plane, the integration domain is this green square, which encompasses $\tilde{\eta}_{A,air,h,square}$.

$$r_{A/B}^{d,B} = DCM\left(\psi_d\right) r_{A/B}^{W} \tag{56}$$

where ψ_d is the angle to rotate F_W so that it aligns with $F_{d,B}$.

Once again, we see that integration of f_B over the $\tilde{\eta}_{A,air,h}$ domain is is difficult to obtain in closed form (this is the integral of a non-symmetric 2D Gaussian distribution over a circular domain of integration that is not centered at the mean). Instead, we may conservatively integrate Eq.54, a square slightly larger than the area of $\tilde{\eta}_{A,air,h}$ to make the calculation simpler. This square region, $\tilde{\eta}_{A,air,h,square}$, (illustrated in Figure 11) is aligned with the $F_{d,B}$ frame and can be expressed as

$$\tilde{\eta}_{A,air,h,square} = \begin{bmatrix} x_1^{d,B} \in \left[r_{A/B}^{d,B}\left(1\right) - R, r_{A/B}^{d,B}\left(1\right) + R\right] \\ x_2^{d,B} \in \left[r_{A/B}^{d,B}\left(2\right) - R, r_{A/B}^{d,B}\left(2\right) + R\right] \end{bmatrix} \tag{57}$$

$\tilde{\eta}_{A,air,h,square}$ is useful because aircraft B's planar position probability distribution function, f_B, can be expressed in its distribution frame, $F_{d,B}$, and can conservatively bound the integral of Eq.54 by the much easier and less complicated integral of

$$p(X_B \in \tilde{\eta}_{A,air,h,square}) = \int\int_{\tilde{\eta}_{A,air,h,square}} f_B^{d,B}\left(x\right) dA \tag{58}$$

where $f_B^{d,B}\left(x\right) = N\left(0, \Sigma_B^{d,B}\right)$

and $\Sigma_B^{d,B} = \begin{bmatrix} \lambda_{B,\max} & 0 \\ 0 & \lambda_{B,\min} \end{bmatrix}$

The closed form expression for this integral was given previously in Eq.40.

Thus, this method conservatively solves an airspace conflict between the perspective entity (aircraft A) and another, non-perspective entity (aircraft B) in the xy-plane.

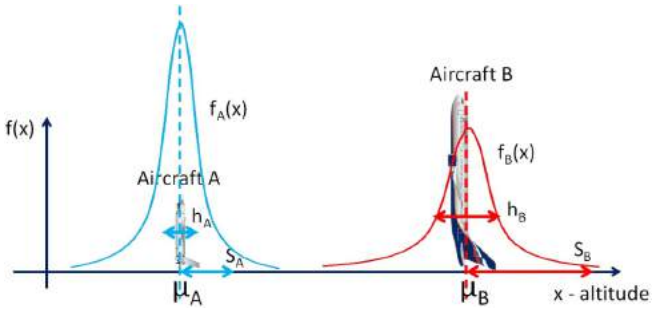

Figure 12. This is a description of the altitude conflict parameters with the perspective entity shown in blue and the non-perspective entity in red.

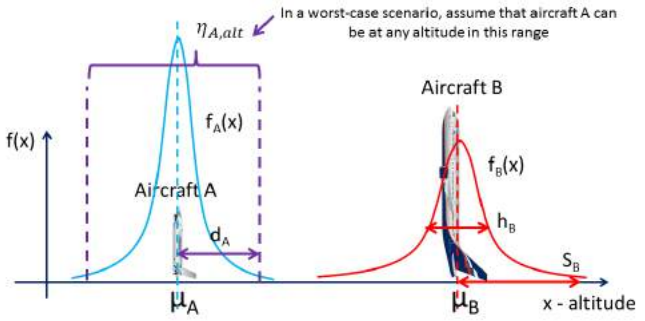

Figure 13. This describes $\eta_{A,alt}$ as the purple range. Note that this is a function of d_A.

Aircraft Conflict Probability in Altitude Direction

Calculating the conflict between two aircraft in the altitude direction follows a similar process. The The following parameters are used to calculate an aircraft conflict probability in the altitude direction. These are also illustrated in Figure 12.

1. h_A is aircraft A's height. This is the actual vertical distance taken up by aircraft A.
2. s_A is aircraft A's vertical airspace separation distance, which is half of the total airspace height.
3. μ_A is aircraft A's mean altitude.
4. σ_A is aircraft A's distribution's standard deviation.

Next the problem can be formulated as an independent probability calculation using the following two random variables, X_A is aircraft A's altitude and X_B is aircraft B's altitude. Since the altitude of aircraft A and aircraft B are independent, the following region is considered, $\eta = [x_{min}, x_{max}]$. The probability that a sample from f_A and f_B falls in this range is given by

$$\begin{aligned} p\left(X_A \in \eta\right) &= \int_{x_{\min}}^{x_{\max}} f_A\left(x\right) dx \\ p\left(X_B \in \eta\right) &= \int_{x_{\min}}^{x_{\max}} f_B\left(x\right) dx \end{aligned} \tag{59}$$

Since the two probabilities are independent, the probability that a sample from both f_A and f_B will be in the range η is given by $p\left(X_A \in \eta \&\& X_B \in \eta\right) = p\left(X_A \in \eta\right) p\left(X_B \in \eta\right)$.

Next a description of aircraft A's airspace needs to be formulated. Because the aircraft's altitude is stochastic, this airspace needs to be conservative and tries to account for regions of high likelihood. Note that it is unreasonable to attempt to account for 100% of the possibilities of aircraft A's altitude because under a Gaussian distribution, this requires the range of $[-\infty, \infty]$.

Instead, consider that aircraft A's altitude can be in a range of

$$\eta_{A,alt} = [\mu_A - d_A, \mu_A + d_A] \tag{60}$$

where $\eta_{A,alt}$ is aircraft A's worst-case altitude range.

To be conservative and assuming a worst-case scenario, aircraft A can be at any altitude in the range of $\eta_{A,alt}$ as described in Figure 13

The probability that aircraft A is located within $\eta_{A,alt}$ is therefore given by

$$\begin{aligned} F\left(d_A\right) &= p\left(X_A \in \eta_{A,alt}\right) \\ &= \int_{\mu_A - d_A}^{\mu_A + d_A} f_A\left(x\right) dx \\ F\left(d_A\right) &= \text{Erf}\left(\frac{d_A}{\sqrt{2}\sigma_A}\right) \end{aligned} \tag{61}$$

This is in effect a type of cumulative probability distribution function which captures the percentage of samples that fall in the range $\eta_{A,alt}$ as d_A varies from 0 to ∞. In practice, a user will specify a desired confidence level, F, and the appropriate distance d_A is calculated to achieve this level of confidence. Therefore, solving Eq.61 for d_A yields

$$d_A = \sqrt{2}\sigma_A \text{Erf}^{-1}\left(F\right) \tag{62}$$

Now that a description of aircraft A's altitude range is obtained. The next step is to describe the airspace that it claims as its own. This is calculated as

$$\eta_{A,air,v} = [\mu_A - \left(d_A + s_A\right), \mu_A + \left(d_A + s_A\right)] \tag{63}$$

where $\eta_{A,air,v}$ is aircraft A's worst-case vertical airspace range and describes the altitude that if any object comes in this range, we call this a vertical airspace violation. From geometry, if aircraft B's altitude is anywhere in the range

$$\tilde{\eta}_{A,air,v} = [\mu_A - H, \mu_A + H] \tag{64}$$

where $H = d_A + s_A + \frac{h_B}{2}$, then aircraft B will breach aircraft A's vertical airspace. This is illustrated in Figure 14.

Finally, the probability that aircraft B will breach aircraft A's vertical airspace can be calculated by using Eq.41 with $\tilde{\eta}_{A,air,v}$ and aircraft B's mean and altitude variance.

Since the horizontal and vertical conflict probabilities are assumed to be independent, the aircraft conflict probability is simply the product between the horizontal and vertical probabilities.

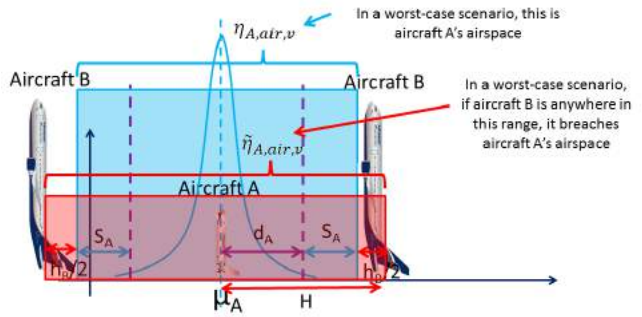

Figure 14. This is a description of the vertical airspace showing $\eta_{A,air,v}$ (the perspective entity's vertical airspace) as the blue range and $\tilde{\eta}_{A,air,v}$ as the range of altitudes that aircraft B must remain outside of in order to maintain conflict free altitudes.

8. Airspace Conflict Calculator

The previous section described algorithms used to determine conservative estimates of conflict between two aircraft. Another problem that UAS operators are concerned with is the conflict between their aircraft (the perspective entity) and restricted airspaces. The airspace conflict calculator provides a conservative probability of an airspace violation. Having this module allows the system to predict if an FSE result will breach an airspace, which is modeled as an extruded non-complex 2D convex or non-convex polygon, where each polygon vertex is known. This system supports only extruded polygons with vertical side walls. In addition to the 2D polygon the height and the airspace's altitude ranges are known. The airspace generates the limits of integration for both the 2D and 1D probability.

Rotating the Airspace

In order to use Eq.40 to calculate the 2D probability a diagonal covariance matrix is required. Once the covariance is rotated using a DCM, a new 2D distribution, centered at the same mean, is generated. Since the covariance is rotated the airspace must also be rotated via the 2D polygon's vertices to maintain the same relative distance between the covariance and the airspace. Figure 15 illustrates how a vertex (Point A) relates to the covariance in both the Cartesian and body frame xy-plane. Let $\left[\begin{array}{cc} x_{vertex}, & y_{vertex} \end{array}\right]^{\mathrm{T}}$ be a polygon's vertex.

A vertex is transformed by

$$\left[\begin{array}{c} \tilde{x}_{vertex} \\ \tilde{y}_{vertex} \end{array}\right] = DCM_{Body/XY}\left(\left[\begin{array}{c} x_{vertex} \\ y_{vertex} \end{array}\right] - \mu\right) + \mu \tag{65}$$

Discretizing the Polygon

After rotating both the 2D Gaussian distribution and the polygon, the polygon is discretized. Discretizing the polygon is the first step necessary to obtain a simplified calculation of the probability of violating the airspace in the xy-plane. The first step in the discretization process is to determine a rectangle which surrounds the polygon. Next this rectangle is discretized based on an operator specified number of blocks in the x-direction and y-direction to create a series of grids within the containing rectangle. The algorithm then determines which grids either touch or are within the polygon. To determine this, the algorithm uses the following condition: if

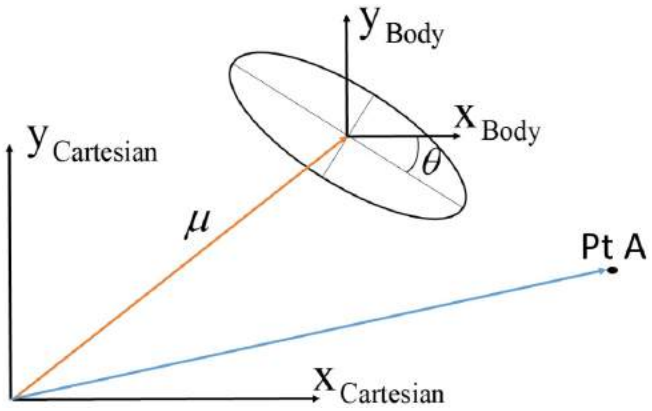

Figure 15. This shows the relationship between the 2D Gaussian distribution and a polygon vertex (Pt A). Furthermore since the Gaussian distribution needs to be rotated to the body frame, Pt A also needs to be rotated to maintain the same relative distance.

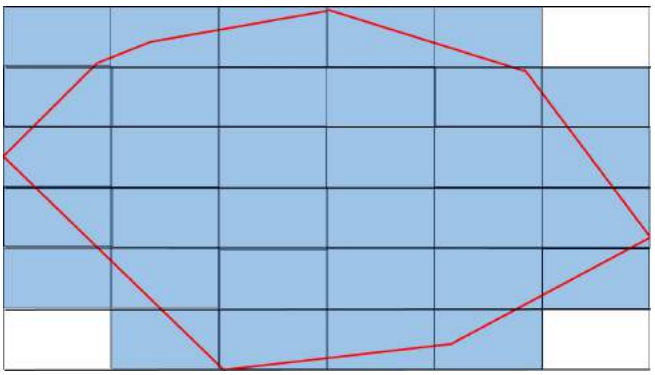

Figure 16. The red polygon represents a 2D airspace. This polygon is then discretized and the blue grids represents the grids that touch or are within the airspace. These blue grids form the polygon grid domain.

any of the four sides of a given grid intersects the polygon, then the grid touches or is within the polygon. Furthermore the algorithm uses a non-optimized winding algorithm to handle cases when the grid is completely within but does not intersect the polygon. Once all the grids that touch or are within the polygon are found, these grids form the polygon grid domain, where N is the number of grids in the polygon grid domain.

Calculating the Airspace Conflict Probability

Since each grid contains a maximum and minimum x-value and y-value, the 2D probability of being in the polygon grid domain can be determined by Eq.40. Let each grid probability be P_{xy_k} where k is the grid number. Thus the probability of being in the polygon grid domain is

$$P_{xy,Tot} = \sum_{k=1}^{N} P_{xy_k} \tag{66}$$

Since the polygon grid domain's area is equal or larger than the polygon's area, this implies $P_{xy,Tot} \geq P_{polygon}$.

Next the altitude probability is calculated using the airspace's

minimum and maximum altitude. Eq.41 calculates P_z. Since $P_{xy,Tot}$ and P_z are independent, the total probability is their product. Also since $P_{xy,Tot}$ is conservative, this implies P_{tot} is also conservative.

$$P_{tot} = P_z P_{xy,Tot} \tag{67}$$

Monte Carlo Simulation

The airspace conflict calculator is validated using a Monte Carlo simulation developed by the research team. 5000 random positions based on the FSE result are generated. Each point is classified as either violating or not violating the airspace. Then the Monte Carlo result is calculated by dividing the number of violations by 5000. Since the Monte Carlo simulation generates random positions, each simulation results in different conflict probabilities. Thus, this Monte Carlo simulation provides an airspace violation approximation.

The following 3 scenarios validate the airspace conflict calculator's conservativeness and the assumption that the probability in the xy-plane and z-direction are independent. A large discrepancy between the Monte Carlo results and the airspace conflict calculator results occurs when the FSE result's covariance is small compared to the airspace and breaches the airspace. In this scenario a discretized airspace may have a grid containing the entire distribution thus the airspace conflict calculator predicts a larger probability than the Monte Carlo result. The airspace conflict calculator predict a larger conflict probability than the Monte Carlo simulation. This is expected because this demonstrates the airspace conflict calculator's conservative nature to overestimate airspace conflicts. Scenario 1 (as shown in Figure 17) illustrates this scenario by having an FSE result approximately halfway breaching the airspace. As expected the Monte Carlo simulation predicts a 48.66% probability (close to the ideal 50% value), while the airspace calculator predicts an expected 100% probability because a discretized polygon grids contains the FSE result's 95% confidence region.

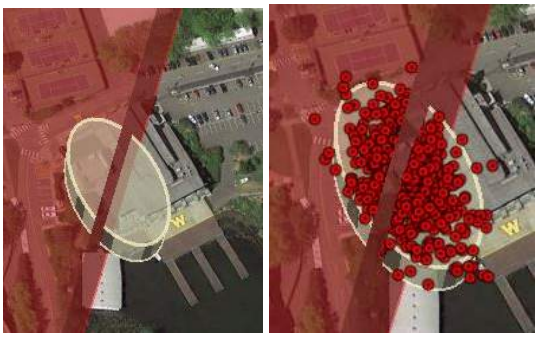

Figure 17. This is scenario 1, where the FSE result halfway breaches the airspace in the xy-plane as illustrated in the left image. The image on the right shows the Monte Carlo result, where the red dots are the random generated positions.

Scenarios 2 and 3 test the independence assumption. Figures 18 and 19 illustrate both the side and top view for scenarios 2 and 3. Note Figure 18 has the side walls removed to show the amount the FSE breaches the airspace. Since in both scenarios the FSE result is contained by the airspace in the xy-plane, the airspace conflict calculator result will vary based on the probability in the altitude direction. Furthermore since the airspace and the FSE result are similar in size, the Monte Carlo simulation and the airspace conflict calculator results should be similar as well.

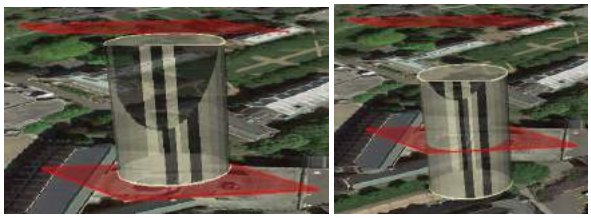

Figure 18. These two images are the side views of scenario 2 (left) and scenario 3 (right). Also the sidewalls of the airspace is removed for clarity. In scenario 2 the FSE result is almost completely inside the airspace in the z-direction. In scenario 3 the FSE result halfway breaches the airspace in the z-direction.

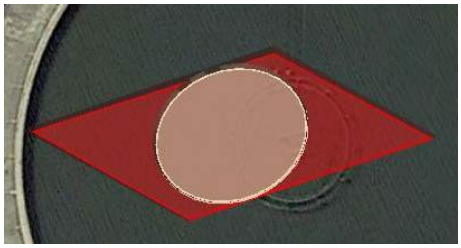

Figure 19. This illustrates the FSE result in the 2D direction for scenarios 2 and 3. For scenarios 2 and 3 the FSE result is contained in the airspace in the xy-plane.

For scenario 2 the Monte Carlo simulation (Figure 20) yields a 96.82% probability, while the airspace conflict calculator predicts a 96.69% probability. For scenario 3 the Monte Carlo simulation (Figure 21) yields a 58.96% probability, while the airspace conflict calculator predicts a 58.56% probability.

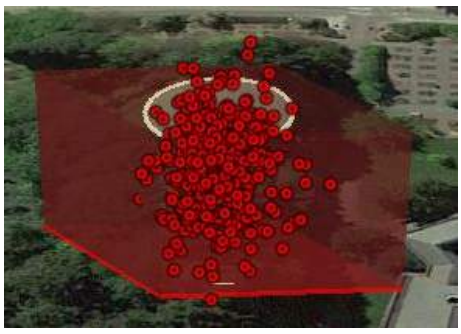

Figure 20. This is the side view of scenario 2's Monte Carlo simulation results. As expected most of the red dots violate the airspace.

Since the airspace conflict calculator's results are either within 2% of the Monte Carlo results or the airspace conflict calculator results are greater than the Monte Carlo results, the aircraft conflict calculator results are conservative and the independence assumptions about the probabilities in the xy-plane and altitude direction is valid.

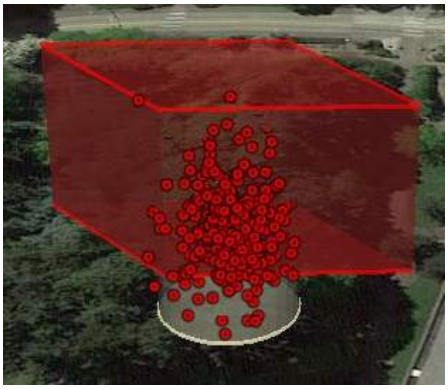

Figure 21. This is the side view of scenario 3's Monte Carlo simulation results. As expected half of the red dots violate the airspace.

9. SIMULATION

Two realistic simulations are engineered to demonstrate the algorithm's capabilities. Scenario 1 highlights the FSE free flight, FSE flight path, aircraft conflict calculator and airspace conflict calculator functionalities, while scenario 2 focuses on the algorithm's FSE free flight, FSE orbit, aircraft conflict calculator and airspace conflict calculator.

Simulation 1

This simulation is 90 seconds in length and contains a restricted airspace and 3 different aircraft. Two are UAS (UAS06 and UAS04) and the third is a passenger jet (Jet01). This scenario exercises the FSE free flight, FSE flight path, and the conflict calculator capabilities. In this scenario the perspective entity is UAS06 and changes to UAS04 halfway in the simulation (simulating the case where the operator starts controlling UAS06 and is then handed off to control UAS04). Initially both Jet01 and UAS04 are in free flight and UAS06 is following a parallel track flight path. Figures 22 and 23 illustrate the initial FSE predictions which include a 30 second prediction time FSE results. At this time, the conflict calculator predicts no potential conflicts.

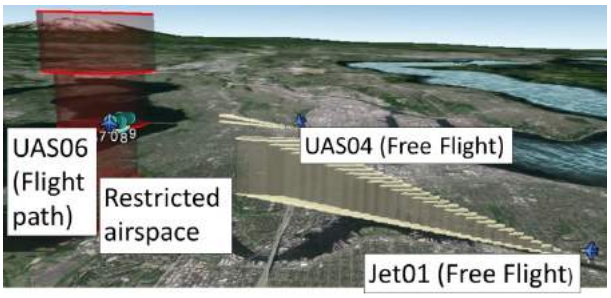

Figure 22. This is a side view of simulation 1's FSE results at 0 seconds. In this scenario, there are 2 aircraft in free flight, 1 aircraft in a flight path mode and an airspace. Forward state estimates are predicted 30 seconds into the future.

In the first 3 seconds there are no potential conflict warnings. However from 4 to 10 seconds, the system predicts Jet01 will be in conflict with UAS06 in the next 30 seconds with 100% probability. Figures 24 and 25 illustrates the FSE result at 10 seconds. Although the two FSE results do not intersect, the conflict calculator predicts 100% probability due to the conservative methods applied.

Figure 23. This is a top view of simulation 1's FSE results at 0 seconds. Furthermore this figure better depicts the FSE results for all 3 aircraft.

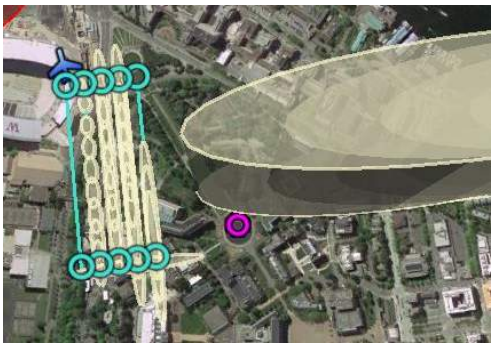

Figure 25. This is a top view of simulation 1's FSE results at 10 seconds, which shows the close proximity between the Jet01's FSE result and UAS06's FSE result.

Figure 24. This is a side view of simulation 1's FSE results at 10 seconds. This illustrates the pending conflict between Jet01 and UAS06.

For the next 14 seconds the system continues to warn of a pending conflict between UAS06 and Jet01 with 100% probability and the time of conflict continues to decrease. At 24 seconds the system predicts the time of this conflict has

decreased to 12 seconds. Since the conflict probability has become too high, the operator decides to change UAS06's flight path from a parallel track to an expanding square flight pattern south of the conflict area. Figure 26 illustrates the FSE results for each aircraft at 25 seconds.

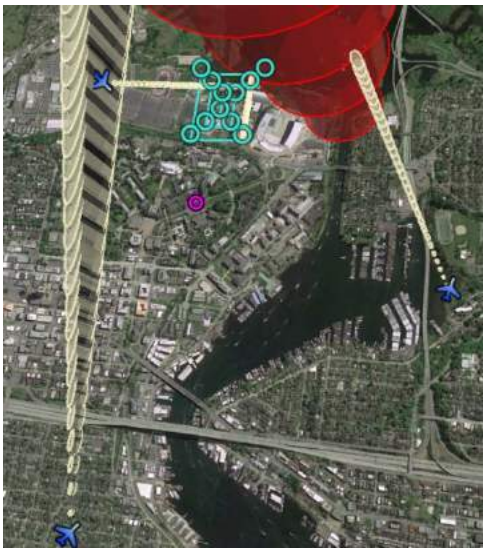

Figure 26. This is simulation 1's FSE results at 25 seconds. At this time since the conflict between UAS06 and Jet01 is highly probable, UAS06's operator changes the flight path from the parallel track to the expanding square.

UAS06 continues to follow the expanding square flight path for the next 20 seconds. At this time UAS04 is still in free flight and Jet01 clears the parallel track as illustrated in Figures 27 and 28. Once the parallel track flight path is cleared, UAS06 returns to the parallel track and the operator changes the perspective entity to UAS04. By changing the perspective entity, the operator gave up command of UAS06 and now commands UAS04. Then the operator changes UAS04's flight mode from free flight to flight path, where UAS04 follows the expanding square flight path.

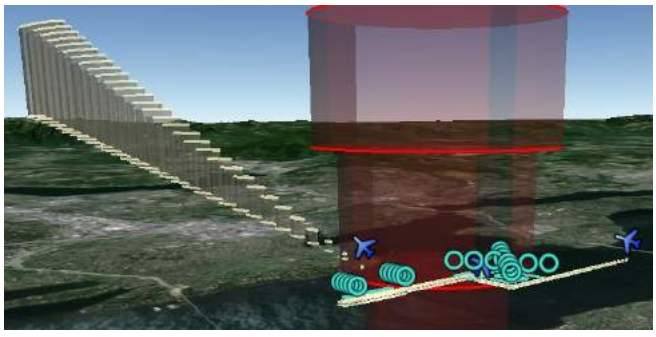

Figure 27. This is a side view of simulation 1's FSE results at 45 seconds. At this time Jet01 clears the parallel track, thus UAS06's operator switches UAS06's flight path back to the original parallel track flight path.

From 51 to 80 seconds the system predicts a conflict between the UAS04 and UAS06 with 100% probability, although both aircraft's FSE results never intersect each other. These warnings are expected because the expanding square and parallel

Figure 28. This is a top view of simulation 1's FSE results at 45 seconds. In addition to switching the flight path of UAS06, the UAS06's operator relinquishes command of UAS06 after sending UAS06 back to the parallel track. Next the operator changes perspective entity and controls UAS04. This operator then switches UAS04's flight mode from free flight to following the expanding square flight path.

track flight route are within UAS04's airspace's horizontal separation distance. Since UAS04's airspace's horizontal separation distance is large and contains UAS06's FSE results, thus the aircraft conflict calculator will predict a 100% conflict probability during this time. However at 81 seconds and for the remainder of the simulation the system warns that UAS04 is currently violating the restricted airspace with 100% probability as illustrated in Figure 29. This 100% probability conflict is expected because the FSE result is contained within an airspace's discretized grid.

Figure 29. This is simulation 1's FSE results at 81 seconds. At this time the airspace conflict calculator determines UAS04 is currently violating the airspace.

Simulation 2

This simulation illustrates the collision awareness algorithm capabilities to handle a general aviation aircraft, (Prop13) operating in free flight mode, as a perspective entity. In addition this simulation spends more time demonstrating the airspace conflict calculator's capabilities. During the simulation Prop13 is in conflict with an orbiting aircraft, UAS05, and a restricted airspace. This simulation is 60 seconds and has a 20 second prediction time. Figures 30 and 31 show the initial FSE predictions. At 0 seconds the conflict calculator predicts UAS05 will breach Prop13's airspace in 14 seconds with 100% probability. As expected the system warns Prop13 about the pending conflict with UAS05 for the first 14 seconds because the FSE results are in close proximity to each other.

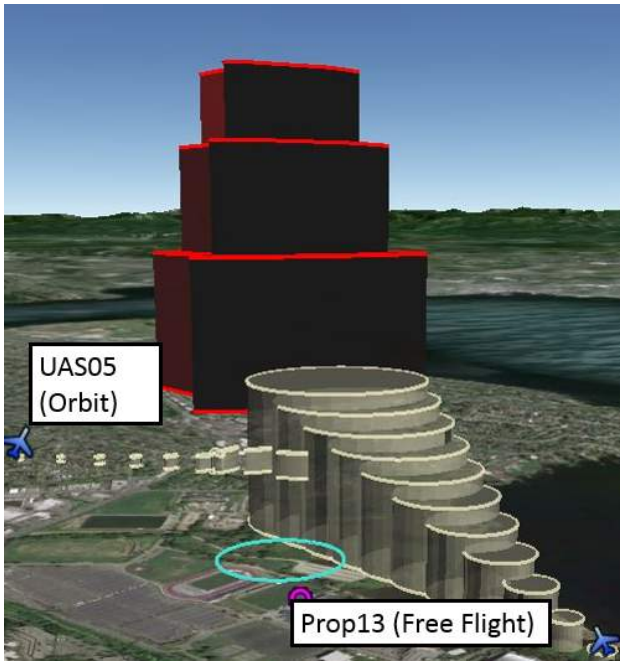

Figure 30. This is a side view of simulation 2's FSE results at 0 seconds. In this scenario, there is 1 aircraft in free flight, 1 aircraft in orbit mode and an airspace.

Figure 31. This is a top view of simulation 2's FSE results at 0 seconds. Furthermore the initial FSE results between UAS05 and Prop13 are within close proximity and the aircraft conflict calculator predicts conflicts between these two aircrafts for the next 14 seconds.

As the simulation progresses the conflict calculator continues to predict that Prop13's airspace will be breached with 100% probability. From 8 to 12 seconds the conflict calculator predicts that UAS05 has breached Prop13's airspace with 100% probability. Once Prop13 passes UAS05's orbit, the system predicts a potential conflict with the airspace as illustrated in Figures 32 and 33. 14 seconds into the simulation the

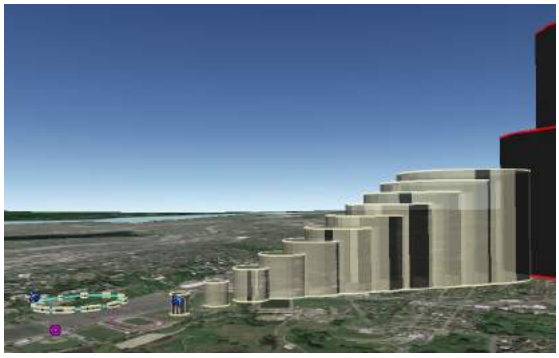

Figure 32. This is a side view of simulation 2's FSE results at 14 seconds. At 14 seconds the warnings between the potential conflicts between Prop13 and UAS05 ceases. The system then warns the operator of a pending airspace violation.

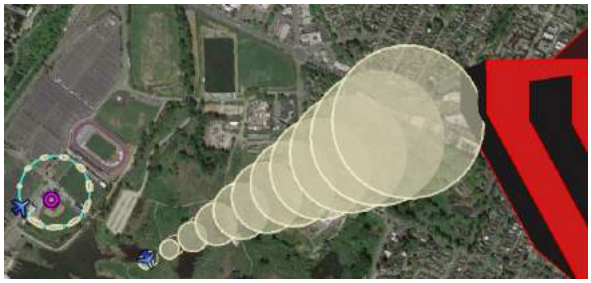

Figure 33. This is a top view of simulation 2's FSE results at 14 seconds. This figure clearly depicts the pending airspace violation by Prop13.

conflict calculator predicts a 6.76% probability of violating the airspace occurring in 10 seconds. As the simulation progresses the conflict calculator continues to predict a violation with increasing probability. For instance at 20 seconds the probability of violation is 61.15% and at 32 seconds the probability of violation is 89.32%.

At 32 seconds the conflict calculator yields an interesting result because the probability of violation is 89.32% in 8 seconds however Figures 34 and 35 show the FSE result does not appear to be contained by 89.32% of the airspace. This discrepancy is caused by the conservative methods implemented in the airspace conflict calculator. Having an over conservative method is important because the pilot would not continue to fly the current heading knowing a high probability of an airspace violation exists.

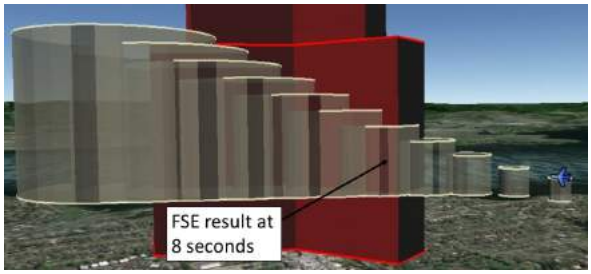

Figure 34. This is a side view of simulation 2's FSE results at 32 seconds. At this time the airspace conflict calculator predicts an 89.32% probability of violating the airspace occurring in the next 8 seconds.

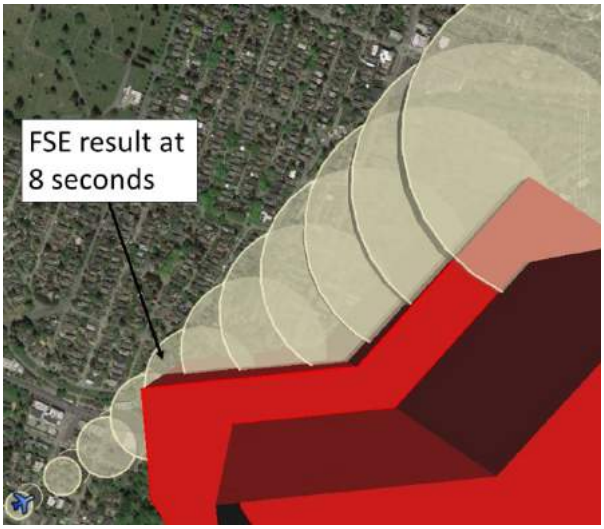

Figure 35. This is a top view of simulation 2's FSE results at 32 seconds. This illustrates the FSE result in the xy-plane and highlights the FSE result at 8 seconds. 8 seconds is important because this is the time the airspace conflict calculator predicts the highest probability of an airspace violation.

Figures 36 and 37 show the FSE results of the aircraft and the airspace at 40 seconds. At this time the conflict calculator predicts that Prop13 is violating the airspace with 100% probability. The conflict calculator continues to predict with a probability close to a 100% that Prop13 is violating the airspace. At 52 seconds Prop13 has cleared the airspace and the conflict calculator has no conflict warnings for the remainder of the simulation.

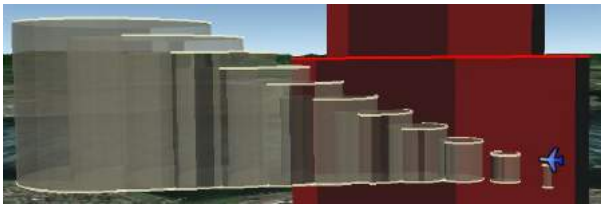

Figure 36. This is a side view of simulation 2's FSE results at 40 seconds. At this time the aircraft is violating the airspace as predicted by the airspace conflict calculator.

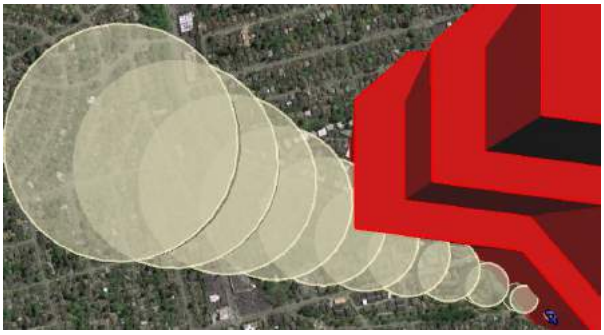

Figure 37. This is a top view of simulation 2's FSE results at 40 seconds. This is the xy-plane view clearly showing Prop13 violating the airspace at 40 seconds.

10. Conclusion

The proposed collision awareness algorithm is simple, overly conservative, and easy to implement. Furthermore, since all equations are in closed form, calculations can be done quickly and requires minimum memory or computational resources. This allows the system to predict positions and conflicts between many aircraft and airspaces at any given moment. By modeling aircraft as 3D point mass, the system can model any UAS or manned aircraft's future position. Although individual aircraft's dynamics are not modeled, by incorporating aircraft specific information into the conflict calculator, vehicle specific predictions are still obtained. Since the system supports three different flight modes and predicts aircraft and airspace conflicts, the system is useful in various scenarios.

Having models that more accurately represents UAS operations when engaged in the flight path and orbit mode will improve the system and provide more realistic conflict probabilities. For both flight modes, the algorithm needs to be modified so that reasonable limits are applied to the σ_{body} growth. Also the orbit mode can be improved by modeling elliptical orbits, thus increasing the different cases the system can handle. Another improvement is modifying the system's prioritizing conflict algorithm. Algorithms discussed in this paper can be incorporated into a warning and notification system which incorporates factors such as the conflict probability, time until conflict, and conflict severity between the aircraft and the other entity to prioritize the conflicts and will greatly increase the operator's situational awareness.

Acknowledgments

The authors would like to recognize the many people and organizations who supported this research project. The authors thank the Washington's Joint Center for Aerospace Technology Innovation (JCATI) for funding this project, the JCATI director, Dr. Mehran Mesbahi, and the JCATI's program manager, Patrick Gibbs. Also the authors are grateful for Insitu's contributions and in-kind support to this project. The authors want to recognize the following Insitu members for their direct involvement with this research project: Dr. Rolf Rysdyk, Keith Ketring, Andrew Hayes, Darcy Davidson, Jim Miller, Aaron High, Amy Arbeit, Jeremy Tate, Jonny Polivka, and Christy Grimm. Also the authors appreciate the following guest speakers for sharing their expertise: Art Crowe, Nathaniel Guy, Matt McCully and Dr. Dick Newman. The authors would like to recognize Laura Dorsey for handling the intellectual property and technology transfer of this research. Lastly, the authors want to acknowledge the research team members who have worked countless hours developing this algorithm: Dr. Juris Vagners, Madison Peck, Richard Fukutome, Matthew Davis, Bao Le, Henry Qin, Brian Chang, Taylor Campbell, Daniel Ablog, Federico Alvarez, Noel Kimber, Dai Tsukada, Robert McSwain, Seunghyun Ko, Ryan Valach, Aleksandr Tereshchenkov, Justin Yantus and John Marshall.

References

[1] "Unmanned aircraft systems (UAS) comprehensive plan: A report on the nation's UAS path forward," Federal Aviation Administration, Tech. Rep., 2013.

[2] "Unmanned systems integrated roadmap," United States Department of Defense, Tech. Rep. 14-S-0553, 2013.

[3] "Integration of civil unmanned aircraft systems (UAS) in the national airspace system (NAS) roadmap, first edition," Federal Aviation Administration, Tech. Rep., 2013.

[4] J. Anno, "Estimate of Human Control Over Mid-Air Collisions," *Journal of Aircraft*, vol. 19, no. 1, pp. 86–88, 1982.

[5] T. McGeer, "Aerosonde hazard estimation," 1994, Aerovel Corporation.

[6] R. A. Clothier, R. A. Walker, N. Fulton, and D. A. Campbell, "A casualty risk analysis for unmanned aerial system (UAS) operations over inhabited areas," in *Proceedings of Twelfth Australian International Aerospace Congress, 2nd Australasian Unmanned Air Vehicles Conference*, Melbourne, March 2007.

[7] C. W. Lum and B. Waggoner, "A risk based paradigm and model for unamnned aerial vehicles in the national airspace," in *Proceedings of the 2011 Infotech@Aerospace Conference*, St. Louis, MO, March 2011.

[8] C. W. Lum, K. Gauksheim, T. Kosel, and T. McGeer, "Assessing and estimating risk of operating unmanned aerial systems in populated areas," in *Proceedings of the 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, Virginia Beach, VA, September 2011.

[9] J. Lai, J. J. Ford, L. Mejias, and P. O'Shea, "Characterization of sky-region morphological-temporal airborne collision detection," *Journal of Field Robotics*, vol. 30, no. 2, pp. 171–193, March 2013. [Online]. Available: http://eprints.qut.edu.au/55883/

[10] T. S. Bruggemann and L. Mejias, "Airborne collision scenario flight tests : impact of angle measurement errors on reactive vision-based avoidance control," in *15th Australian International Aerospace Congress (AIAC15)*, B. Falzon, Ed., Melbourne, VIC, February 2013. [Online]. Available: http://eprints.qut.edu.au/57810/

[11] Federal Aviation Administration. (2014) Fact sheet - unmanned aircraft systems (UAS). [Online]. Available: https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=14153

[12] C. W. Lum, J. Vagners, and R. T. Rysdyk, "Search algorithm for teams of heterogeneous agents with coverage guarantees," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 7, pp. 1–31, January 2010.

[13] C. W. Lum and J. Vagners, "A modular algorithm for exhaustive map searching using occupancy based maps," in *Proceedings of the 2009 Infotech@Aerospace Conference*, Seattle, WA, April 2009.

[14] C. W. Lum, J. Vagners, J. S. Jang, and J. Vian, "Partitioned searching and deconfliction: Analysis and flight tests," in *Proceedings of the 2010 American Control Conference*, Baltimore, MD, June 2010.

[15] C. W. Lum and R. T. Rysdyk, "Time constrained randomized path planning using spatial networks," in *Proceedings of the 2008 American Control Conference*, Seattle, WA, June 2008.

[16] C. W. Lum, M. Vavrina, J. Vagners, and J. Vian, "Formation flight of swarms of autonomous vehicles in obstructed environments using vector field navigation," in *Proceedings of the 2012 International Conference on Unmanned Aircraft Systems*, June 2012.

[17] C. W. Lum, M. L. Rowland, and R. T. Rysdyk, "Human-in-the-loop distributed simulation and validation of strategic autonomous algorithms," in *Proceedings of the 2008 Aerodynamic Measurement Technology and Ground Testing Conference*, Seattle, WA, June 2008.

[18] M. L. Cummings, A. Kirschbaum, A. Sulmistras, and J. Platts, "Stanag 4586 human supervisory control implications," *Air and Weapon Systems Department, Dstl Farnborough & the Office of Naval Research*, 2006.

[19] Federal Aviation Administration, "Introduction to TCAS II," FAA, Tech. Rep., February 2011, version 7.1.

[20] M. McCully, "Interview with a retired air traffic controller," August 2013.

[21] Insitu. (2013) ICOMC2. [Online]. Available: http://www.insitu.com/systems/icomc2

[22] M. Peralta, *Propagation of Errors: How to Mathematically Predict Measurement Errors*. CreateSpace, 2013.

BIOGRAPHY

Kevin Ueunten *received his B.S. in Astronautical Engineering and Mathematical Science from the US Air Force Academy in 2013. He is currently working on his M.S. in Aeronautics & Astronautics at the University of Washington. He is a research member of the Autonomous Flight Systems Laboratory. His current research activities and interests include unmanned aerial systems, collision awareness algorithms, and software unit testing.*

Christopher Lum *received his PhD in Aeronautics & Astronautics from the University of Washington in 2009. He is currently a research scientist at the University of Washington's William E. Boeing Aeronautics & Astronautics Department and runs the Autonomous Flight Systems Laboratory. His research interests includes coordinated multi-vehicle searching, automatic target recognition, formation flight of swarms of vehicles, risk assessment of UAS in the national airspace, collision avoidance/deconfliction, and UAS flight operations. He has worked closely with industry partners such as the Boeing Company, Insitu, Aerovel, and the Washington Joint Center for Aerospace Technology Innovation to implement academic technologies onto deployed platforms. Dr. Lum teaches both undergraduate and graduate course on automatic control, flight mechanics, modeling and simulation, mathematical tools for engineers, sensors and actuators, and other controls related courses. He has been awarded the department's "Instructor of the Year" award twice (2012 and 2013). He is also the faculty advisor to the department's design, build, fly team. He has served as an adjunct professor at Seattle University and as a visiting fellow at the Queensland University of Technology in Brisbane, Australia.*

Al Creigh *is currently an undergraduate student studying Aeronautics & Astronautics at the University of Washington. He is a research member of the Autonomous Flight Systems Laboratory. In 2013 he interned at Applewhite Aero and helped develop, test and certify their unmanned aerial systems. His current research activities include developing, constructing and flying unmanned aerial systems. He is the primary flight test pilot for the University of Washington UAS flight testing program.*

Keisuke Tsujita *is currently an undergraduate student studying Aeronautics & Astronautics at the University of Washington. He is a research member of the Autonomous Flight Systems Laboratory. He received the Robert Max Reynolds Endowed Scholarship Awarded for his academic performance. In 2013 he worked as an intern at Panasonic's Automotive & Industrial Systems Company. His current research activities include unmanned aerial system sensors and image processing algorithms.*