Collision Avoidance Strategies for Cooperative Unmanned Aircraft Systems using Vehicle-to-Vehicle Communications

Jaya Sravani Mandapaka
Electrical Engineering
University of North Texas
Denton, TX, United States
Email: JayaSravaniMandapaka@my.unt.edu

Batool Dalloul
Electrical Engineering
University of North Texas
Denton, TX, United States
Email: BatoolDalloul@my.unt.edu

Skyler Hawkins
Electrical Engineering
University of North Texas
Denton, TX, United States
Email: SkylerHawkins@my.unt.edu

Kamesh Namuduri
Electrical Engineering
University of North Texas
Denton, TX, United States
Email: Kamesh.Namuduri@unt.edu

Shane Nicoll
Unmanned Experts Inc.
Glendale, CO, United States
Email: s.nicoll@unmannedexperts.com

Keven Gambold
Unmanned Experts Inc.
Glendale, CO, United States
Email: kagambold@unmannedexperts.com

***Abstract*—A massive increase is expected in the use of Unmanned Aircraft Systems (UAS) for civilian applications in the near future. Such high-density traffic, especially in urban settings, leads to an elevated risk of collisions in civilian airspaces. Existing collision avoidance solutions that are primarily developed for manned aircraft are not portable to autonomous vehicles and are not sufficiently capable of handling high-volume traffic. Standards organizations such as IEEE, Radio Technical Commission for Aeronautics, and General Aviation Manufacturers Association are working towards developing collision avoidance strategies for cooperative UASs based on Vehicle-to-Vehicle (V2V) communications. This paper presents collision avoidance methods for cooperative UASs in structured and unstructured airspaces using V2V communications. It highlights a use-case scenario "Crossroads," to demonstrate the proposed collision avoidance strategies. Results obtained through simulations carried out using AirSim are also presented.**

***Index Terms*—Unmanned Aircraft Systems, Collision Avoidance Systems, Vehicle-to-Vehicle Communications**

I. Introduction

The current generation of Unmanned Aircraft Systems (UAS) Traffic Management (UTM) typically oversees small UAS flying below 400 ft. in uncontrolled airspace. Although UAS transportation services are currently limited, the industry for commercial UAS and UAS-related services is projected to grow exponentially in the near future due to the Advanced Air Mobility (AAM) campaign currently being pioneered by the National Aeronautics and Space Administration (NASA). AAM aims to develop a nation-wide air transportation system for both people and cargo at altitudes ranging from 500 to 3000 ft. Due to the probable increase in high-density UAS traffic in and around urban areas, there is also an increased likelihood of Mid-Air Collisions (MAC) and Near Mid-Air Collisions (NMAC) over these areas.

Considering AAM operations are set to take place in controlled airspace, there is a need for Collision Avoidance (CA) systems that can provide real-time resolutions to traffic conflicts involving both manned and unmanned aircraft. The current generation of CA systems can be classified into one of these categories: Manned aircraft collision avoidance systems, Detect-and-Avoid (DAA) systems, and hybrid collision avoidance systems.

A. Manned Aircraft Collision Avoidance Systems

The primary types of CA systems for manned aircraft are:

1) **Airborne Collision Avoidance System (ACAS)**: ACAS is a short-range CA system that prevents MAC independent of a centralized base station.
2) **Airborne Separation Assurance System (ASAS)**: ASAS is a long-range CA system that aims to maintain a standard horizontal separation distance of 5 nautical miles (9.3 Km) and a vertical distance of 1000 ft. (300m) between aircraft. [1].
3) **Traffic Collision Avoidance System (TCAS)**: TCAS is a Federal Aviation Administration (FAA) mandated CA system that actively monitors the altitude and distance of other aircraft equipped with Mode C and Mode S transponders. If there is a risk of MAC between aircraft, this system will provide traffic resolution advice to the pilots by issuing alerts that inform them to either climb or descend.
4) **Portable Collision Avoidance System (PCAS)**: PCAS is a CA system similar to TCAS, with the caveat being that it doesn't actively interrogate the transponders of other aircraft; rather, this system actively listens for nearby transponder-equipped aircraft and notifies pilots if any are detected.

979-8-3503-1114-3/23/$31.00 ©2023 IEEE

Unfortunately, CA systems, designed specifically for manned aircraft, are not conducive to AAM operations for several reasons. Firstly, they have almost exclusively been developed for pairwise collisions – meaning that they do not function well in areas experiencing high-density traffic. Second, the size and weight of these systems make them untenable for small UASs, as they can take up to a sq. ft. of space and weigh up to 20 pounds.

B. Detect-and-Avoid Systems

DAA systems are generally based on the concept of using a unique suite of sensors to gather data about the local environment in order to calculate safe trajectories that avoid collisions with intrusive objects. In previous works, partially observable Markov decision process, Markov decision process, and Monte Carlo simulations are used to define state-space and action-space such that the generated trajectory path is collision-free [2] [3]. These types of systems are also referred to as Obstacle Avoidance (OA) systems. The types of sensors typically used in these systems are:

1) **Stereo Vision (SV)**: SV allows UASs to extract information about their environment through the use of one or more stereoscopic cameras [4].
2) **Laser Imaging, Detection, and Ranging (LiDAR)**: LiDAR is used to detect obstacles with pulsed lasers of visible or ultraviolet light. This method offers highly precise measurements in both indoor and outdoor environments. [5]
3) **Infrared Sensors (IR)**: IR sensors use pulsed lasers of infrared light to detect obstacles. Similar to LiDAR, this method also provides excellent results in both indoor and outdoor environments [6].
4) **Ultrasonic Sensors (US)**: US sensors detect obstacle distances by transmitting ultrasonic waves and measuring the time it takes for the waves to reflect off the obstacle surface and return to the sensor [7].

As opposed to the CA systems for manned aircraft, DAA systems ultimately fall short in terms of functionality if implemented for the purpose of avoiding collisions with other aircraft. This is due to numerous factors, with the most notable being that the amount of data that must be processed at any given moment drastically limits the aircraft's maximum speed. Therefore, for the purposes of AAM applications – where moderate to high-speed travel is to be regularly expected – this category of CA systems must be disqualified.

C. Hybrid Collision Avoidance Systems

Hybrid CA systems are those that can be deployed on both manned and unmanned aircraft, effectively bridging the gap between them. Due to platform cross-compatibility, these systems represent the bleeding edge of CA systems for the purposes of AAM operations.

1) **Airborne Collision Avoidance System X (ACASX)**: ACASX is a CA system similar to TCAS in that it detects other equipped aircraft using onboard surveillance systems [8]. This system is classified into six sub-types, each of which can communicate with the other sub-types to offer resolution advice in case there is a risk of a MAC between aircraft. ACAS Xa is for commercial aircraft, ACAS Xu is for unmanned aviation, ACAS Xo is for specific operations such as military applications, and ACAS Xp is for general aviation. ACAS Xr is for rotor aircraft, and ACAS SXu is for small UAS. These ACAS X systems not only avoid a collision but also provide a vertical resolution; but, for unmanned aircraft, both vertical and horizontal resolutions are needed.
2) **Vehicle-to-Vehicle Communication Systems (V2V)**: V2V Communication systems enable aircraft to communicate with each other through a decentralized network. This system allows aircraft to effectively articulate their intentions to other nearby aircraft, while simultaneously monitoring their positions and velocities. If there is a risk of a MAC between two aircraft, resolutions to the conflict are generated and autonomously executed.

Naturally, Hybrid CA systems are prime candidates for AAM operations due to the remarkable advantages they have over the aforementioned systems. Specifically, the V2V system stands out for its ability to keep track of the positions, velocities, and intentions of other nearby aircraft [9]. However, even with its advantages, V2V communication systems are still not a universal solution to the problem at hand. This is due to the fact that – in the context of V2V communications – an aircraft can be one of two types: cooperative or non-cooperative [10]. Cooperative aircraft are those that are capable of communicating with one another through a decentralized network, while non-cooperative aircraft are those that are not [11]. For these reasons, the focus of this paper is solely on CA systems for cooperative aircraft using V2V communications.

II. Research Problem

As explained in the introduction, NASA's AAM initiative has sparked widespread interest in the development of UAS-related technologies and infrastructure. To accommodate the next generation of UAS technologies, CA systems must be developed that can support a high-density UAS ecosystem. A plausible candidate to fill this role is the V2V communication system, which has steadily been gaining traction in research institutions around the nation in recent years. This paper aims to demonstrate the capabilities of these systems in practical scenarios.

A. Main Contributions

The main contribution of this paper is presenting effective CA strategies for cooperative vehicles equipped with V2V communication systems. The strategies discussed in this paper are applicable in both structured and unstructured airspace.

1) **Structured Airspace** is airspace where the flights should travel in designated pathways or tracks. This is often seen in an urban environment where the use of airspace is limited due to tall buildings and the density of vehicles is high. A flight trip of $<$ 50 km can travel in structured airspace. Air corridors [12] and UAS

take-off/landing sites, known as vertiports, are prime examples of structured airspace.

2) **Unstructured Airspace** is airspace where there is no specified path or tracks designed for flights to travel which is often called free flight mode. This is seen in rural areas. These types of airspaces can be used for long-distance flights. A manned aircraft traveling internationally uses this kind of airspace.

Additionally, explanations are given as to what the necessary components are for a cooperative CA system, along with descriptions of how those components are meant to be used. To accomplish this, the concept of "Crossroads" is introduced, which is an aerial four-way intersection that mimics an equivalent ground-based intersection. Subsequently, the concepts of prioritized and non-prioritized aircraft are introduced, along with the accompanying concepts of yielding and Right of Way (ROW). To demonstrate the capabilities of the CA strategies being proposed, the Crossroads scenario is modeled in simulation using Microsoft's AirSim simulator.

B. Organization of the Paper

The organization of this paper is as follows: section II presents the problem that this paper aims to solve with V2V-based cooperative CA systems. Section III details the concept of Crossroads and the necessary components of V2V message exchanges. Section IV explains the use of the PX4 autopilot along with the implementation of predefined and user-defined flight modes. Section V outlines the method of utilizing V2V messages to facilitate the traversal of various intersection types within structured and unstructured airspace. Section VI covers the simulations that are performed along with the results that are obtained. Lastly, section VII concludes the paper with a summary of findings and details regarding upcoming future works.

III. CROSSROADS

Crossroads is a special use-case scenario conceptualized for developing a cooperative V2V collision avoidance system. The idea behind Crossroads is to have two vehicles safely navigate through an intersection by means of a V2V message exchange. Within this scenario, each of the two vehicles is assigned a priority level which dictates how the vehicle will react if a collision is detected. The vehicle priority levels are currently being arbitrarily assigned as either one (prioritized) or zero (non-prioritized). If the priority level of one vehicle is higher than the other, then the vehicle with the highest priority gets ROW through the intersection, and the other vehicle must yield. The method in which a non-prioritized vehicle yields is directly linked to the vehicle's active user-defined flight mode and other variables. Further explanation of predefined and user-defined flight modes is provided in Section IV. Several vehicle parameters and functions facilitate the CA process, namely the vehicle heartbeats, parlance radius, bump radius, Minimum Safe Radius (MSR), and the delta envelope.

A. Vehicle Functions/Parameters

*1) **Heartbeats**:* A heartbeat is the basic form of communication between vehicles. Heartbeats are continuously being broadcast containing information such as the vehicle's telemetry, speed, heading, etc.

*2) **Parlance Radius**:* A vehicle's parlance radius is where communications with other vehicles initiate. This radius is a user-defined value that allows a vehicle to accept and recognize the heartbeats being sent from another vehicle. Fig.1 illustrates the parlance radius.

*3) **Bump Radius**:* The bump radius is the distance from another vehicle at which immediate action should be taken to avoid a collision. It should be noted that the bump radius must always be less than the parlance radius. Fig.1 also illustrates the bump radius.

*4) **Minimum Safe Radius**:* The MSR is the distance between vehicles that must be maintained to be considered safe. The MSR can be either static or dynamic depending on the application; however, the ideal implementation uses a dynamic MSR directly proportional to the vehicle's current velocity.

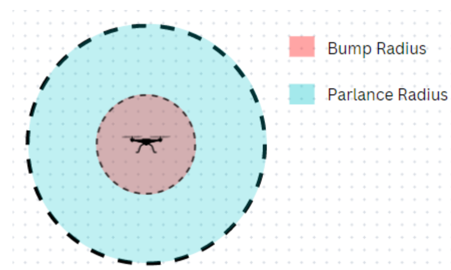

Fig. 1. Parlance Radius and Bump Radius

*5) **Delta Envelope**:* A vehicle's delta envelope describes the properties that define its quantifiable limitations during flight based on its physical characteristics and capabilities [13]. These limitations include properties such as the minimum/maximum velocities, accelerations, and load factors. Fig. 2 shows the maximum/minimum velocities denoted as β_{max} and β_{min}, with the current velocity denoted as v. The variables δ_+ and δ_- indicate the difference between the maximum/minimum velocities and the current velocity.

B. Components of Intersection

*1) **Intersection Point**:* Once a vehicle detects another vehicle in its parlance radius, they must begin negotiations as to which drone has the highest priority. To find out which vehicle has the highest priority, the vehicles analyze the heartbeats of the other vehicle and compare them to their own. Once this has been determined, the vehicle with the highest priority calculates the intersection point based on the overlap of both vehicle trajectories. Vehicle trajectories are modeled by taking the vehicle heading and projecting it outwards as a 3D vector. In aviation, these vectors are referred to as "great circles," which mathematically represent circles on the earth's surface that lie on a plane passing through the center of the earth [14]. To calculate a vehicle's great circle vector, the vehicle's current

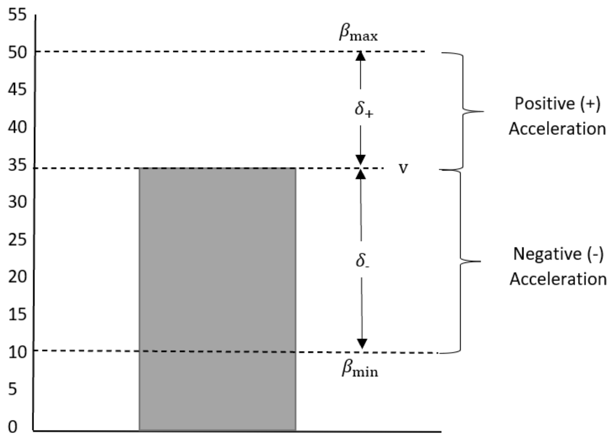

Fig. 2. Delta Envelope Diagram

longitude (ϕ), latitude (λ), and heading (θ) (with respect to the north) are required. The equations used to perform these calculations are detailed below [14] [15].

$$\vec{c} = \begin{bmatrix} \sin(\phi)\cos(\theta) - \sin(\lambda)\cos(\phi)\sin(\theta) \\ -\cos(\phi)\cos(\theta) - \sin(\lambda)\sin(\phi)\sin(\theta) \\ \cos(\lambda)\sin(\theta) \end{bmatrix} \tag{1}$$

For finding the correct intersection point, it requires taking the cross product of the trajectory vectors of both vehicles, hence $\vec{c_1} \times \vec{c_2}$, and $\vec{c_2} \times \vec{c_1}$. Great circles always intersect at exactly two points, meaning the correct intersection point must be identified from the pair. The method to determine which intersection point is valid is to check the distance between the points and the vehicles and determine if both vehicles are headed toward that point. In some cases, when the vehicles enter each other's parlance radius, the intersection point can render itself behind one of the vehicles. In this situation, there is no danger of collision, and the intersection point is disregarded.

*2) **Estimated Time of Arrival**:* The Estimated Time of Arrival (ETA) is the approximate time it takes for a vehicle to reach the intersection point if it maintains its current velocity. It is worth noting that the distance (d) used in calculating a vehicle's ETA is not the displacement but rather the distance the flight path covers leading up to the intersection point.

$$t_{ETA} = \frac{d}{v} \tag{2}$$

*3) **Designated Crossing Time**:* The Designated Crossing Time (DCT) consists of two parts - the upper and lower DCT. The lower DCT is generated by calculating the time at which the vehicle MSR first comes into contact with the intersection point, whereas the upper DCT is generated by the time at which the vehicle's MSR separates contact with the intersection point. In equation 3, r_{MSR} indicates the vehicle's current MSR, while v indicates the vehicle's current velocity.

$$t_{DCT} = (t_{ETA} - \frac{r_{MSR}}{v},\ t_{ETA} + \frac{r_{MSR}}{v}) \tag{3}$$

*4) **Intersection Message**:* The intersection message is the message that contains the information required to allow a non-prioritized vehicle to yield to a prioritized vehicle. This message is sent by the prioritized vehicle to the non-prioritized vehicle as an extension of the heartbeats transmitted at regular intervals. The contents of the intersection message include the intersection point, ETA, DCT, and distance to the intersection. Once these values have been calculated by the prioritized vehicle, it appends them to the heartbeats being sent to the non-prioritized vehicle. On the receiving end, the non-prioritized vehicle parses this information from the heartbeats and uses it to calculate a solution that allows it to yield to the prioritized vehicle safely. The type of solution that is calculated directly results from the active user-defined flight mode and type of airspace the vehicles are flying in.

IV. Flight Modes

A. PX4 Flight Modes

There are several different flight modes available for vehicles running PX4; however, most of them are not relevant for the purposes of this paper, with the exceptions being PX4's offboard mode and mission mode [16]. When attempting to avoid a MAC through V2V communications, offboard mode has thus far proved to be the premier solution due to its malleable development framework and options for vehicle control. During simulation, this mode only becomes active once an imminent collision between two vehicles is detected; otherwise, the vehicle remains in PX4's Mission Mode.

*1) **Offboard Mode**:* Offboard mode gives the onboard companion computer access to the vehicle's directional controls, including direct commands for the vehicle's velocity, position, and attitude (pitch, roll, yaw controls). This offers many degrees of freedom and allows for the execution of unorthodox vehicle movement patterns; however, there is sometimes difficulty in using offboard mode because there is only a limited set of commands currently supported. Subsequently, another limiting factor is that only one type of vehicle control can be used at once; that is, a vehicle can only be controlled by its velocity, position, or attitude but not by any combination of the three at once.

*2) **Mission Mode**:* Mission mode is used for autonomous vehicles to execute a user-defined flight plan. This is the default mode a vehicle operates in for both structured and unstructured airspace until an imminent collision is detected.

B. User-Defined Flight Modes

For this series of experiments, a vehicle engaging in V2V communications must always operate in one of the aforementioned PX4 flight modes and one user-defined flight mode. User-defined flight modes differ from the PX4 flight modes in that they are a set of operational parameters implemented in software running on the vehicle's companion computer, separately from the vehicle's flight controller firmware (PX4). The user-defined flight modes currently include free-flying mode and air corridor mode, which have unique characteristics.

*1) **Free-Flying Mode:*** Free-flying mode represents any sort of autonomous and unstructured flying that doesn't take place within structured airspace. This mode is characterized as dynamic, meaning way-points may be updated intermittently while the vehicle is in flight.

*2) **Air-Corridor Mode:*** Air corridor mode differs from the free-flying mode in that vehicles are only permitted to fly within precisely defined air corridors. An air corridor is a type of structured airspace with strict regulations regarding the minimum/maximum altitude, the direction of travel, and takeoff/landing locations a vehicle may use when operating in that airspace [12].

V. Collision Avoidance Strategies

Thus far, the necessary components for resolving intersection conflicts via V2V communications have been defined. There are currently three types of intersections that have been conceptualized using the intersection components. Out of these, two types are for free-flying mode, which includes altitude-yield and velocity-yield intersections. On the other hand, for air corridor mode, the roundabout intersection is used.

A. Unstructured Airspace - Free Flying

The CA process for both the altitude-yield and velocity-yield intersections generally follow the same paradigm. The process begins when two vehicles enter each other's parlance radius, at which point they begin actively analyzing the contents of each other's heartbeats and monitoring each other's positions/velocities. As the vehicles continue converging, they eventually enter each other's bump radius, which prompts the prioritized vehicle to assess whether there is an intersection point between them.

If an intersection point is detected between the two vehicles, the prioritized vehicle records the location of this point and begins generating the intersection message. The prioritized vehicle's ETA and DCT are also attached to this message before it is appended to the next outgoing heartbeat and sent to the non-prioritized vehicle. It should be noted that the intersection message is continuously sent out with each subsequent heartbeat until the conflict is been resolved, in an effort to mitigate any issues caused by packet loss during transmission. At this point, the non-prioritized vehicle receives the intersection message and analyzes its contents. Afterwards, the vehicles resolve the intersection conflict using one of the two following methods:

*1) **Altitude-Yield Intersection:*** The altitude-yield intersection allows two vehicles to avoid an intersection conflict by means of an altitude adjustment in one of the vehicles. Once the non-prioritized vehicle receives a heartbeat with an intersection message attached, it is prompted to resolve the intersection conflict by adjusting its altitude. To calculate this trajectory, an algorithm is used that returns a list of position set points in the shape of a cosine wave with the midpoint lying directly over or under the intersection point.

This algorithm uses a numerical approximation technique to determine points on the curve which are of equal arc lengths (L) away from one another. The drop distance (β) is the height at which the drone will fly over or under the other drone. The action distance (α) is the distance at which the drone begins ascending/descending. The point b_n is the point on the curve which is L distance away from the starting point on the curve (a_n). The conditions for the algorithm are presented below:

$$\text{while } \mathrm{a}_n < \alpha : \begin{cases} n = 0 : a_0 = 0 \\ n \geq 1 : a_n = b_{n-1} \end{cases} \quad (4)$$

The following equation is used to solve for the series of b_n values that satisfy the conditions above.

$$L = \frac{1}{2}(b_n - a_n)\left(\sqrt{1 + \left[\frac{-\beta\pi\sin\left(\frac{\pi a_n}{\alpha}\right)}{2\alpha}\right]^2} + \sqrt{1 + \left[\frac{-\beta\pi\sin\left(\frac{\pi b_n}{\alpha}\right)}{2\alpha}\right]^2}\,\right) \quad (5)$$

The list of generated (a_n, b_n) pairs is then converted from Cartesian coordinates into WGS-84 coordinates using a series of geometric transformations. First, the points are rotated around the z-axis by an angle equal to the drone's heading from north (θ).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_n \\ 0 \\ b_n \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} (6)$$

Afterwards, a scaling factor based on the current location of the drone is applied to the points that converts them into WGS-84 coordinates. Subsequently, the latitude, longitude, and altitude of the intersection point are superimposed on top of the scaled points and added together. Note that the longitude scaling factor uses the latitude point of the intersection (λ_{int}) during calculations [17] [18].

$$\text{Longitude Scaling Factor} : \frac{1}{111111\cos(\lambda_{int})} \quad (7)$$

$$\text{Latitude Scaling Factor} : \frac{1}{111111} \quad (8)$$

Once the list of position set points is generated, the non-prioritized vehicle switches into PX4's offboard mode and follows the generated trajectory above the other vehicle. After crossing the intersection successfully, it switches from off-board mode back into PX4's Mission mode and continues its original mission.

*2) **Velocity-Yield Intersection**:* The velocity-yield intersection is a simple way for two vehicles to avoid an intersection conflict through means of a velocity adjustment in one of the vehicles. Once the non-prioritized vehicle receives a heartbeat with an intersection message attached, it is prompted that action needs to be taken in order to resolve the intersection conflict by calculating a solution velocity.

The solution velocity is the velocity that a non-prioritized vehicle must yield to a velocity-yield intersection to avoid a collision. This velocity must satisfy certain constraints as dictated by the delta envelope of the yielding vehicle. The constraints imposed on the calculated solution are such that the final velocity must not exceed or fall below the yielding aircraft's maximum or minimum velocity; also, it must not require an acceleration for which the vehicle is incapable.

To calculate the solution velocity (v_f), the yielding vehicle uses its own distance from the intersection (d), initial velocity (v_i), and desired duration (t_f). The duration is chosen to be the same time as the prioritized vehicle's lower DCT, since this allows the yielding vehicle to cross the intersection as soon as it has been cleared by the prioritized aircraft. The equations used for these calculations are derived from an ideal constant-jerk trajectory generator, which returns the solution velocity and the accompanying acceleration rate needed to achieve this velocity.

The first step is to find the constant (a_c) and maximum accelerations (a_{max}) that are required by the trajectory.

$$a_c = \frac{2d - 2v_i t_f}{t_f^2} \text{ and } a_{max} = 2a_c \tag{9}$$

Next, the event's target velocity (v_f) is calculated.

$$v_f = \sqrt{v_i^2 + 2a_c d} \tag{10}$$

After finding these values, they must be compared to the yielding vehicle's delta envelope to ensure that none exceed or fall below the established minimums and maximums, respectively. If a valid solution is found, the non-prioritized vehicle switches to PX4's offboard mode and uses velocity controls to yield to this velocity at the calculated acceleration rate, allowing the prioritized vehicle to cross the intersection. Afterwards, the non-prioritized vehicle switches back to PX4's Mission mode and resumes the mission item that it was previously executing.

B. Structured Airspace- Air corridors

*1) **Roundabout Intersection**:* At a high level, the properties of the roundabout are mathematically defined using two common equations, namely the circle and equilateral hyperbola. This idea can be conceptualized by graphing these equations – which share a relationship based on the circle's radius – in a 2D Cartesian plane, essentially representing an overhead view of the roundabout if it is projected on the xy-plane in 3D space. In a similar way used in the altitude-yield intersection, position set-points are calculated using an equation analogous to that in equation (5). The difference is that instead of deriving points from a cosine wave, these set points are derived from the equations of a circle and hyperbola. Based on the route identified in the flight plan, the proper WGS-84 coordinates are taken from these segments and uploaded to the vehicle before the flight begins. Once the route has been fully integrated into the vehicle's flight plan, the vehicle can effectively communicate its intentions.

VI. Simulations

The simulator used for Crossroads is Microsoft's AirSim. AirSim is a free development platform available as a plugin for Unreal Engine 4 for individuals, students, and researchers alike. This simulator was chosen over counterparts like JMavsim and Gazebo because of the visual fidelity and unique features such as custom environments and multi-vehicle simulations. In addition, the PX4 autopilot software is also used in conjunction with AirSim as Software-In-The-Loop (SITL) to simulate two UAS using Raspberry Pis. The operating system used on the Raspberry Pis is Ubuntu 18.04, which is recommended by the PX4 developers. All drones in the simulations are programmed to transmit heartbeats at 2Hz, have a parlance radius of 150 meters, and a bump radius of 30 meters.

A. Velocity-Yield

The velocity-yield intersection has been extensively tested in simulation and has performed well thus far. In Fig.3, the delta envelope is configured to have a minimum/maximum velocity of 5-20 m/s, meaning it can't drop below 5 m/s or go above 20 m/s with the calculated solution velocity. At the beginning of the simulation, it is considered that both vehicles travel towards the intersection point at 5 m/s. The non-prioritized vehicle (pictured left) is prompted to yield to the upper DCT of the prioritized vehicle (pictured right). After both vehicles cross the intersection, they successfully resume their original missions.

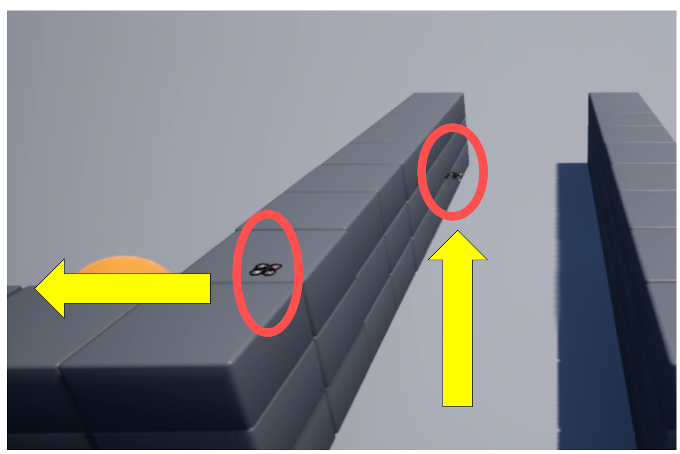

Fig. 3. Velocity-Yield Intersection

B. Altitude-Yield

Like the velocity-yield intersection, the altitude-yield intersection has been tested in simulation with promising results.

Conversely, the altitude-yield solution does not rely on the constraints of the non-prioritized vehicle's delta envelope because it is performed using an altitude adjustment instead of a velocity adjustment. In Fig.4 the non-prioritized vehicle (pictured center) starts to increase its altitude once prompted by the prioritized vehicle (pictured right). After the non-prioritized vehicle crosses the intersection, it returns to its original altitude and resumes its mission.

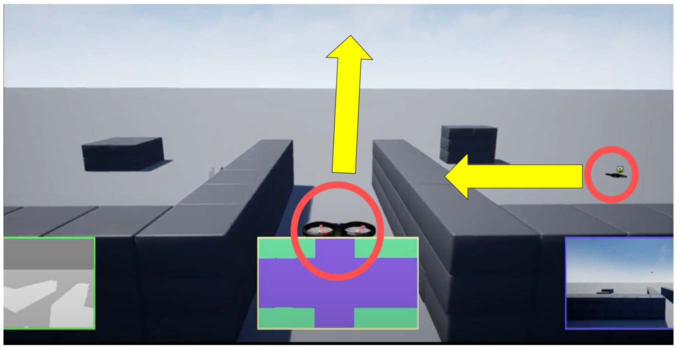

Fig. 4. Altitude-Yield Intersection

VII. Conclusions

In conclusion, the methods in this paper have been presented for resolving a MAC between two vehicles. The simulation results indicate that the velocity-yield and altitude-yield intersections are well-fitted for unstructured airspace. Once refined to a further degree, they can be effectively implemented in real-world CA systems. V2V communications are still in an infantile stage, and the best practices for implementing systems of this type have not been entirely established. System modularity is bound to play a critical role in unmanned CA systems. This is one of the driving factors in developing this system; hence, the necessary components for resolving intersection conflicts in free-flying and air corridor modes are being defined for use in multiple intersection types. System modularity is bound to play a critical role in unmanned CA systems. Therefore, it is beneficial to all parties if a predefined set of protocols is used in these scenarios.

Overall, the obtained results thus far suggest a prospective future for unmanned CA systems. Due to the evolution of the UAS ecosystem and the level of prominence this area of study has garnered in academia, future works will be made to include yielding solutions tailored for use in structured airspaces to satisfy the demands of industry leaders working towards vehicle travel in air corridors. Although the roundabout intersection is mentioned in the previous sections, this method is still under development and requires more time to work out the elements required for full implementation. However, good progress is being made on this front. In addition, research is being conducted to find methods to adapt the previously discussed solutions to include scenarios involving more than two vehicles, along with solutions that allow for multiple layers of yielding vehicles through a single intersection. Methods such as these are yet to be developed; however, the concepts put forward in this paper are making such ideas a reality.

Acknowledgements

This material is based upon work supported by the National Science Foundation under grant CNS -2148178 (Resilient & Intelligent NextG Systems) as well as the NASA Collaborative Research Opportunity (AAM-NC). Support provided by the National Aeronautics and Space Administration, Advanced Air Mobility National Campaign, North Texas Cohort, and Unmanned Experts is appreciated.

References

[1] Quan Quan, Rao Fu, and Kai-Yuan Cai. How far two uavs should be subject to communication uncertainties. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[2] Weizhi Feng, Cheng-Chao Huang, Andrea Turrini, and Yong Li. Modelling and implementation of unmanned aircraft collision avoidance. In *International Symposium on Dependable Software Engineering: Theories, Tools, and Applications*, pages 52–69. Springer, 2020.

[3] Peng Zhao, Weichang Wang, Lei Ying, Banavar Sridhar, and Yongming Liu. Online multiple-aircraft collision avoidance method. *Journal of Guidance, Control, and Dynamics*, 43(8):1456–1472, 2020.

[4] Jongho Park and Hoki Baek. Stereo vision based obstacle collision avoidance for a quadrotor using ellipsoidal bounding box and hierarchical clustering. *Aerospace Science and Technology*, 103:105882, 2020.

[5] Subramanian Ramasamy, Roberto Sabatini, Alessandro Gardi, and Jing Liu. Lidar obstacle warning and avoidance system for unmanned aerial vehicle sense-and-avoid. *Aerospace Science and Technology*, 55:344–358, 2016.

[6] S Sabikan, SW Nawawi, and S Sudin. A survey of onboard sensors for quadrotor collision avoidance system. *J. Eng. Appl. Sci*, 12:4138–4143, 2017.

[7] Kevin N Cobankiat, Timothy A Pua, Rembrandt V Que, Willis Jordan Y Wee Ebol, and Alvin Y Chua. Quadrotor research platform with obstacle avoidance. In *Applied Mechanics and Materials*, volume 842, pages 283–287. Trans Tech Publ, 2016.

[8] John Deaton and Michael P Owen. Evaluating collision avoidance for small uas using acas x. In *AIAA Scitech 2020 Forum*, page 0488, 2020.

[9] Kota Kondo, Reinaldo Figueroa, Juan Rached, Jesus Tordesillas, Parker C Lusk, and Jonathan P How. Robust mader: Decentralized multiagent trajectory planner robust to communication delay in dynamic environments. *arXiv preprint arXiv:2303.06222*, 2023.

[10] Andrew Weinert, Luis Alvarez, Michael Owen, and Benjamin Zintak. Near midair collision analog for drones based on unmitigated collision risk. *Journal of Air Transportation*, 30(2):37–48, 2022.

[11] Josh Bertram and Peng Wei. Distributed computational guidance for high-density urban air mobility with cooperative and non-cooperative collision avoidance. In *AIAA Scitech 2020 Forum*, page 1371, 2020.

[12] Sabrina Islam Muna, Srijita Mukherjee, Kamesh Namuduri, Marc Compere, Mustafa Ilhan Akbas, Péter Molnár, and Ravichandran Subramanian. Air corridors: Concept, design, simulation, and rules of engagement. *Sensors*, 21(22):7536, 2021.

[13] Aerospace News. What is flight envelope in aviation. http://monroeaerospace.com/blog/what-is-flight-envelope-in-aviation, 2021.

[14] Movable Type Scripts. Vector-based geodesy. http://www.movable-type.co.uk/scripts/latlong-vectors.html#intersection, 2021.

[15] Steven P Burns and Jeff J Scherock. Lambert guidance routine designed to match position and velocity of ballistic target. *Journal of guidance, control, and dynamics*, 27(6):989–996, 2004.

[16] PX4. Px4 flight modes overview. https://docs.px4.io/main/en/getting_started/flight_modes.html, 2021.

[17] Howard Veregin. How big is a degree? https://www.sco.wisc.edu/2022/01/21/how-big-is-a-degree/, 2022. [Online; accessed 18-April-2023].

[18] John P. Snyder. Map projections: A working manual. 2012.