Detect and Avoid (DAA) with Broadcast Remote ID (RID)

Aashay S. Shah, Wentao Chen, Jason Ho, Samuel Jawhar, Tiantian Lu, Mengjie Xie, Pengfei Yan, Peter J. Burke *Fellow, IEEE*

***Abstract*—We design, implement and demonstrate a detect and avoid (DAA) system for unmanned aerial vehicles (UAVs) using broadcast Remote ID (RID). An on-board RID receiver detects UAVs and automatically, autonomously (with no pilot input during the flight) takes avoidance action if a user-defined perimeter is breached (in the xy position and z altitude). On breach, the UAV will land, return to launch (RTL), move horizontally, or move vertically. A video alarm is displayed on the on-screen display (OSD), as well as on ground control stations (GCS) connected via wireless telemetry. All of the software and designs are open source, freely available to the drone community, and can be implemented with off-the-shelf parts. This paper demonstrates one possible detect and avoid technology option for collision avoidance between UAVs.**

***Index Terms*—Remote Identification (Remote ID, RID), detect and avoid (DAA), unmanned aerial vehicles (UAVs), unmanned aircraft systems (UAS), collision avoidance, anti-collision.**

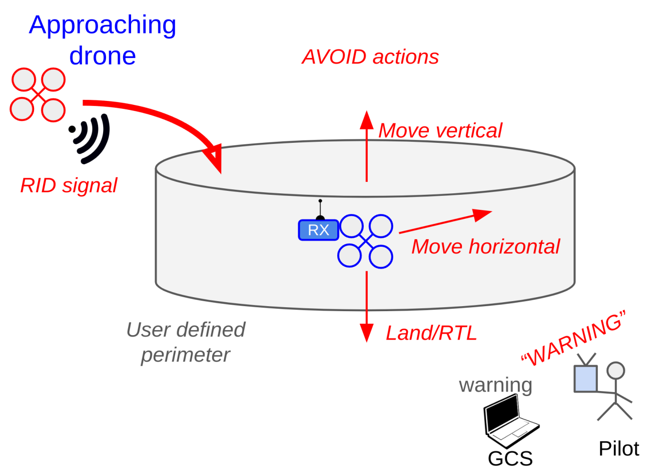

Fig. 1. Concept of operation (CONOPS). An RID receiver is integrated into the center drone, which detects any nearby drones within RID range. If the user-defined perimeter ("bubble" or "hockey puck") is breached, the drone takes avoidance action. The remote pilot is alerted to any drones in range on the on-screen display (OSD) and ground control station (GCS).

I. Introduction

One of the most fundamental, unsolved challenges with introducing drones into the shared airspace is collision avoidance. With the advent of the drone era, collision avoidance is paramount for future integration into a country's airspace system. The number of registered UAV pilots in the USA is approaching one million and continues to grow, and already vastly exceeds the number of registered manned pilots [1]. To date, the vast majority of operations are line-of-sight single-pilot operations, which limits the technological promise of drones from evolving to their full potential.

Collision avoidance between manned aircraft is widely based on the presence of the pilot in the aircraft, as well as automated technologies such as ADSB transponders and radar. However, for UAVs, for operations beyond visual line of sight to be possible, automated approaches are by definition the only option for reliable and safe collision avoidance technology. The FAA recently released its draft beyond visual line-of-sight rules [2] (BVLOS). Those draft rules *require* detect and avoid for permission to fly BVLOS. Both ADSB and radar have been extensively investigated for drone collision avoidance [3]–[10]. (LIDAR [11]–[15] and FLARM [16]–[18] have also been used.) However, ADSB out is only for manned aircraft and is forbidden for drones. Radar detection of drones is very challenging due to the low speed and radar cross section. Therefore, due to the laws of physics and the legal restriction of ADSB for manned aircraft, the only feasible method for drones is some sort of drone-based active transmitter/transponder architecture.

Wentao Chen, Jason Ho, Samuel Jawhar, Tiantian Lu, Aashay S. Shah, Mengjie Xie, Pengfei Yan, Peter J. Burke are with the Department of Electrical Engineering and Computer Science, University of California, Irvine, as well as BME, MSE, CBEMS, Physics (Burke). e-mail: pburke@uci.edu

Manuscript received February, 2025.

As of this writing (summer 2025), the transponder architecture envisioned by the drone industry is called remote id (RID). The end goal for unmanned traffic management is to have any drone on the internet during flight, called network remote ID [19] . However, that is not yet required legally and is not yet in wide use technologically. In addition, a cloud-based management system needs a backup in case of disconnect. A simpler version called broadcast remote ID is in broad use and legally mandated in many countries under a variety of operating conditions. As of this writing (Spring 2024), all drone pilots in the United States are required to equip and operate their drones with Remote Identification (RID) to transmit identification and location information while in flight under the enforcement of the Federal Aviation Administration (FAA) [20]. Europe and Asia have similar requirements. Data such as drone real-time location/altitude, drone serial number, operator ID/location, etc. are broadcast via Wi-Fi or Bluetooth from the drone. Although it was mandated primarily for the identification on the ground of drones in the air, it was also believed that it could be used to avoid collisions and was mentioned in passing in the official ASTM standard [21].

Several recent works have taken this idea and simulated what the appropriate parameters would be for a realistic RID-based collision avoidance system in the air [22]–[24]. However, to date, no broadcast RID collision avoidance system has been developed in software or demonstrated in hardware, in the field, for drone-to-drone collision avoidance.

Here we develop, build, demonstrate, and validate a complete hardware/software drone-drone detect-and-avoid (DAA) collision avoidance stack based on broadcast remote ID. We demonstrate this proof-of-concept complete DAA system using a custom drone with off-the-shelf RID receivers and other hardware, and code implementation integrated into the flight controller as an autonomous, robust, lightweight, and self-contained on-board DAA system. The collision avoidance is completely autonomous and requires no pilot input during the flight, providing real-time DAA with instant, automatic collision avoidance response once a threat is detected. Our system, which is open source and only costs a few dollars to implement on most drone systems, provides an important tool in the DAA toolbox needed for safe integration of drones into the airspace system.

II. Concept of Operation (CONOPS)

Fig. 1 shows the concept of operation. The receiving drone has an RID receiver that receives all broadcast signals in range. The drone takes avoidance action if a user-defined perimeter is breached (in the position xy and z altitude). The drone can be configured with a predefined action to land, return to launch (RTL), move horizontally, or move vertically. A video alarm is displayed on the on-screen display. A ground control station can also be configured to signal an alarm and provide real-time telemetry of avoidance action and the detected drone location.

III. System Design

The design is composed of a custom-made UAV for demonstration purposes with custom software described next in detail.

A. Hardware design

The hardware and avionics are described here.

1) Bill of materials: The bill of materials (BOM) is listed in Table I. The total cost of all the parts was around 500 US dollars in Spring 2024.

2) Assembled product: The fully assembled drone is shown in Fig. 2.

3) Avionics: The avionics are shown in Fig. 3. The flight controller (an STM32-based microcontroller MATEKSYS F405-WMN running custom firmware described below) receives RID data of the detected drone over a UART port. The RID receiver is an ESP32 development board, model ESP32-C3-DevKitM-1. The ESP32 microcontroller has built-in Wi-Fi and Bluetooth capabilities. For the purposes of this demonstration, the Bluetooth receiver was used, but both could in principle be programmed to detect broadcast RID on both protocols. The ESP32 RID receiver was mounted on the top of the drone as shown in Fig. 2. The frame, ESCs, motors, etc. are standard off the shelf parts for DIY drones, and listed in detail here: https://rotorbuilds.com/build/31747. This airframe was

TABLE I
Bill of Materials

Purpose	*Part*
Flight Controller (FC)	MATEKSYS F405-WMN
Motor	T-Motor F1404 3800Kv Micro Motor
GPS	BN880
RX Radio Receiver	2.4GHz EP2 ELRS Nano Receiver
Telemetry	Seeed Studio XIAO ESP32C3
Frame	MOD-L 3-3.5-4-5 inch : modular ultralight frame (with separate arms)
Props	Gemfan Hurricane 4024 Durable Bi-Blade 4” Prop
Camera and VTX	AIO cam Wolfwhoop WT07 Micro 5.8GHz 25mW FPV Transmitter and 600TVL Camera
ESC	Spedix ES20 Lite 2-4S 20A ESC
Battery	2S LiPo
Video Receiver	FYS 40CH 5.8GHz Diversity FPV Monitor w/ DVR - 4.3”
Radio Controller (RC)	RADIOMASTER POCKET M2 RC TRANSMITTER ELRS

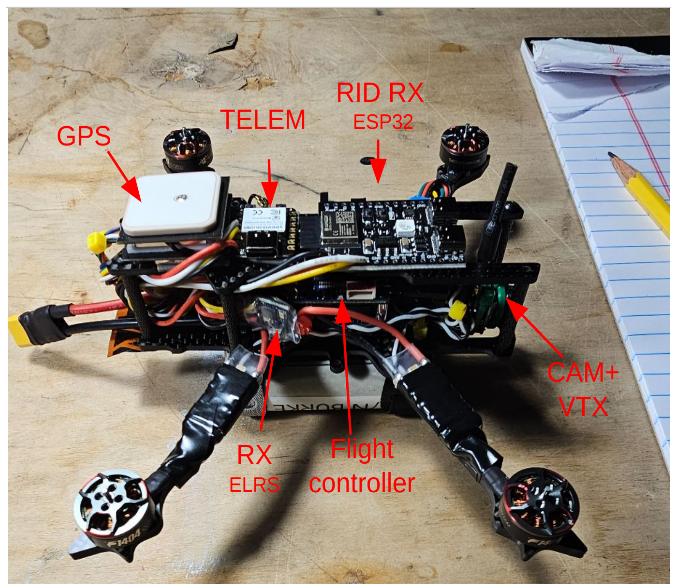

Fig. 2. Fully assembled drone used in this work. The RID receiver is shown mounted on the top of the drone.

also part of a widely acclaimed class (EECS 195 Drones), one of the authors taught in fall 2024, where around 80 students built and flew this exact model drone.

We used a relatively new Wi-Fi-based air-to-ground telemetry link [25], based on another ESP32 board, to maintain a MAVLink interface between the drone and a ground control station, in this case an instance of Mission Planner on a Windows 10 laptop.

B. Software design

The overall design of the software is discussed next.

1) RID receiver: For the RID receiver, custom firmware was developed [26]. The receiver firmware uses Open Drone ID to parse received Bluetooth packets. Open Drone ID is an open source implementation of RID. Once parsed, the receiver

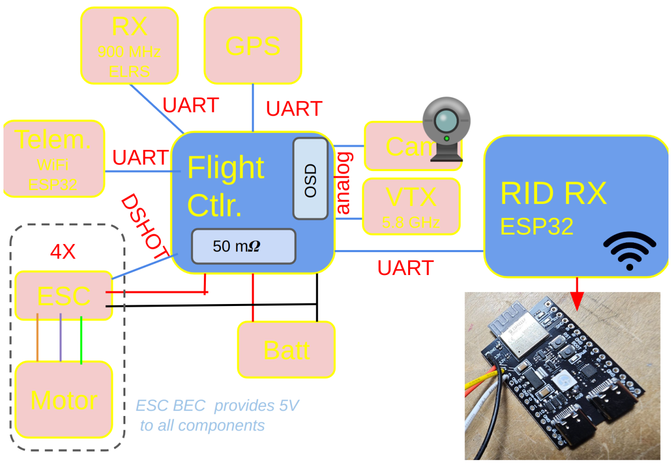

Fig. 3. Avionics. The RID receiver listens for broadcast signals from all drones in range, and sends the data to the flight controller over UART using the MAVLink protocol. A picture of the ESP32 board used is shown, with the Bluetooth PCB antenna clearly visible at top left.

converts the packets to MAVLink packets and sends them over UART to the flight controller.

2) Flight controller: For the flight controller, a custom version of ArduPilot was developed [27]. The custom version receives the RID data on UART over MAVLink and then compares the detected drone location to a user-defined perimeter set in the parameters on the flight controller. When the approaching drone breaches this perimeter, the flight controller puts the drone into one of four possible actions: RTL, land, move horizontally, or move vertically. When the approaching drone leaves this perimeter, the drone returns to its original state or continues in its new state, depending on the setting defined by the user.

In addition, the flight controller displays the distances xy and z to the nearest approaching drone on the screen display (OSD) through the video transmitter to alert the pilot, and a warning is displayed at the ground control stations (GCS).

C. Flight logs and post flight analysis

1) Flight logs of UCIRID drone: For the UCIRID drone (the drone with collision avoidance), flight information was recorded/logged in three different formats. Onboard the drone flash memory, a detailed log was saved in binary format for offline download after the flight. During flight, the GCS is also in constant communication with the drone using MAVLink telemetry, and recorded log files locally on the PC in *tlog* [28] format. Finally, the OSD and the video feed were recorded on an SD card in the video receiver for further post-flight analysis. Mission Planner software was used for postflight analysis of log files.

It should be noted that the standard for first person view (FPV) analog cameras used has limited resolution. It is based on NTSC and PAL. PAL has a resolution of 720×576 at 25 fps (frames per second), while NTSC has 720×480 at 30 fps. In particular, due to the low resolution, it is difficult or impossible to see the approaching DJI drone in the FPV video. In fact, that is one of the points of this paper, which is autonomous collision avoidance via radio detection that does not require manual, visual observation by the pilot either with his naked eyes or the FPV video feed.

The post-flight analysis of the UCIRID drone was performed with three different software tools: Desktop software Mission Planner, MAVExplorer, or cloud-based log file analysis plot.ardupilot.org. All three could plot each parameter and RID imminent collision warnings and autonomous avoid action messages, as well as flight mode changes, vs time, and also plot the drone location on a 2d or 3d map, as well as export KML files for further analysis in additional mapping software such as Google Earth.

2) Flight logs of DJI drone: A DJI drone was used as the threat drone, with an external RID module attached. The DJI drone logs its data into a proprietary on-board log file, which can be uploaded to a commercial cloud log file analysis provider (Airdata UAV, www.airdata.com) and parsed via a web browser. Due to the proprietary nature of the log files, only limited information is available, such as plot on a map.

Finally, the DJI on board camera can record high-quality 4k video. For some of the demonstration flights, we used the 3x zoom to record the UCIRID drone from the DJI drone perspective. However, even with the high-quality 4k camera, it was still very difficult to see the UCIRID drone in the recorded camera footage. The real-time footage visible to the pilot on the ground is even lower resolution (1080 p). Because of this, even with a 4k camera and 3x zoom, visual detect and avoid would be extremely challenging to implement for the general use case. Therefore, the radio detect and avoid presented in this work is still significant and advantageous over visual or optical camera methods.

IV. COLLISION AVOIDANCE ALGORITHM

The algorithm demonstrated in this paper is a simple perimeter breach algorithm. When the UAV detects a potential collision, it *automatically* takes evasive action, without any pilot input during the flight. Thus, it is a completely autonomous collision avoidance strategy. During the flight, the decision is automated and the DAA strategy is fully autonomous. Prior to the flight, the pilot provides input on which avoidance maneuver the drone will take if a collision is calculated as imminent by the on board processor.

A. Pre-flight planning

Before the flight, the pilot makes a conscious, manual (not automated) decision on which of the following avoidance maneuvers the drone will take during flight if a collision is determined as imminent:

- Move horizontally
- Move vertically
- Land
- Return to launch (RTL)

Furthermore, the pilot also decides in advance of the flight the size of the collision bubble, specifically the following parameters:

- Horizontal bubble size

- Vertical bubble size

Before the flight, the pilot also chooses which of the following actions the drone automatically takes after the avoidance maneuver, once the danger has passed:

- Loiter (stay in place)
- Continue in the flight mode it was in prior to the collision
- Land
- RTL

All three decisions are entered as parameters and uploaded to the drone flight controller prior to the flight. Once the flight commences, the pilot has no input unless manual override is engaged. All avoidance procedures are fully autonomous during the flight. This is the entire purpose of this work: The pilot input is not needed during the flight, eliminating the need for fast manual response by a human, thus increasing safety and reliability.

In this work, the decision process for selecting the appropriate avoidance maneuver and bubble size is manual. Our goal here was to demonstrate all possible avoidance maneuvers. However, in general, the ideal decision will depend on the anticipated flight mission profile. For example, land is the most conservative and safest maneuver in most cases, but it terminates the flight. In the future, more research is needed on automating the decision process to select the appropriate avoidance maneuver, based on the flight profile.

B. During flight collision avoidance algorithm

In this work, the avoidance decision is based on the detection of a breach of a user-defined bubble. The pilot can select the perimeter size on a flight-by-flight basis prior to the flight. The user-defined input parameters (which can be set in Mission Planner and uploaded to drone memory after the code is compiled flight by flight) are xy, z (Fig. 1) and Action. Once the perimeter is breached, based on the velocity of the approaching drone, the software (autonomously independent of pilot input during the flight) makes the decision to take avoidance action.

The actual puck / bubble size that should be used for "well-clear" (safe separation) has been quantified for the avoidance of manned aircraft - UAV collisions by the extensive work of Weinert and colleagues [29]–[35]. He found that 2000 feet horizontal and 500 feet vertical was the appropriate hockey puck for well clear. However, no such consensus standard for the appropriate size of the hockey puck for drone-drone collision avoidance has been established so far. Weinert discusses progress towards this in [36], where he finds (Table 3) 15-20 m horizontal, and 5 m vertical for the appropriate bubble size for drone-drone mid-air collision (MAC) avoidance. Well clear bubbles are usually taken to be larger than MAC bubbles. For example, in Ref. [37] (Tables 4,5) recommend 60-85 m as the bubble size for well-clear drone-drone collisions. Therefore, we chose a bubble size of between 10 and 50 m horizontal and 10 m vertical.

This bubble-based collision avoidance algorithm is suitable for slow-moving approaching drones. It is not appropriate for high-speed approaching drones, where avoidance should be initiated long prior to the breach of the predefined perimeter. That would have to be a topic for future research. The exact speed at which this approach will fail is not investigated in this paper, but a rough guess is based on the RID standard. The RID broadcast standard, similar to the ADSB standard, requires a position broadcast at around one Hz. Therefore, if the approaching drone is moving so fast that the collision risk is substantial for the pilot selected perimeter, then a more complicated collision avoidance algorithm will need to be developed. For example, for a perimeter of 10 m, a speed of greater than 10 m/s (22 mph) of the approaching drone would be too fast for this method. If the perimeter is 100 m, a speed greater than 100 m/s (220 mph) would be too fast for this method, etc.

C. Mathematical avoidance algorithm

If move horizontal action is selected as the parameter by the user, then the drone will execute one of two possible maneuvers: First, if the approaching zone speed is less than 1 m/s (at the moment hard coded into the firmware, although this could be a user-defined parameter in the future), the drone will simply fly directly away from it. The reasoning here is that, if the approaching drone is slow, flying directly away will always take the drones further apart.

On the other hand, if the approaching drone speed is greater than 1 m / s, then the algorithm that is used is shown in Fig. 4. The formula for calculating V_{avoid} (the blue arrow in Fig. 4 is as follows:

$$\begin{aligned}
\vec{p}_1 &= \text{Obstacle position (in NED)} \\
\vec{p}_2 &= \text{Ownship position (in NED)} \\
\vec{v}_1 &= \text{Obstacle velocity (in NED)} \\
\Delta\vec{p} &= \vec{p}_2 - \vec{p}_1 \\
\vec{r} &= \Delta\vec{p} - \left(\frac{\Delta\vec{p}\cdot\vec{v}_1}{\|\vec{v}_1\|^2}\right)\vec{v}_1 \\
\vec{r}_\perp &= \frac{\vec{r}}{\|\vec{r}\|} \quad \text{(normalized perpendicular vector)}
\end{aligned}$$

and

$$\begin{aligned}
\vec{v}_{\text{avoid}} &= \text{normalize}\left(\Delta\vec{p} - \left(\frac{\Delta\vec{p}\cdot\vec{v}_1}{\|\vec{v}_1\|^2}\right)\vec{v}_1\right) \\
&= \frac{\Delta\vec{p} - \left(\frac{\Delta\vec{p}\cdot\vec{v}_1}{\|\vec{v}_1\|^2}\right)\vec{v}_1}{\left\|\Delta\vec{p} - \left(\frac{\Delta\vec{p}\cdot\vec{v}_1}{\|\vec{v}_1\|^2}\right)\vec{v}_1\right\|}
\end{aligned}$$

The avoidance algorithm computes an optimal evasion direction based on the relative position and velocity of the intruding aircraft. Specifically, given the obstacle's position and velocity vectors, and the ownship's position, the algorithm first determines the relative displacement vector between the two aircraft. It then projects this displacement vector onto the plane perpendicular to the obstacle's velocity vector. This projection yields a direction that maximizes lateral separation from the trajectory of the incoming aircraft. By normalizing this perpendicular component, the algorithm produces a unit vector representing the optimal avoidance direction. This approach ensures that evasive maneuvers prioritize maximizing

This article has been accepted for publication in IEEE Aerospace and Electronic Systems Magazine. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MAES.2026.3668143

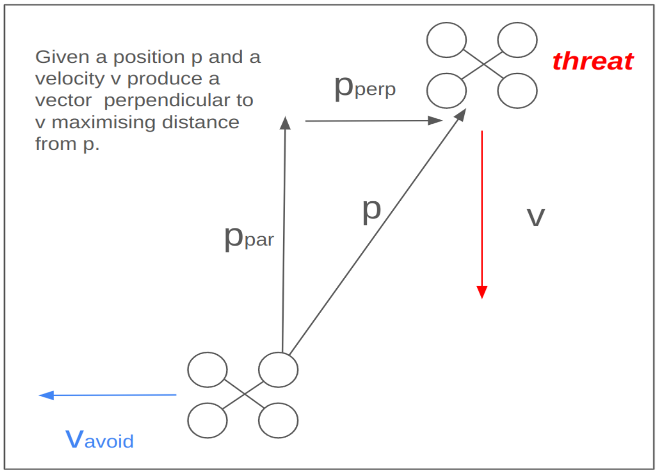

Fig. 4. Collision avoidance geometry.

the rate of separation from the predicted path of the intruder, reducing the likelihood of collision.

The position and velocity of the offending UAV is encoded in the RID packet. This is required according to FAA and ASTM standards [20], [21]. This is received by the RID receiver and used by the on-board flight computer. This is the red ”v” in Figure 4 The position and velocity of the UAV itself are determined by the GPS onboard.

In summary, this formula produces a vector to maximize the distance from p. Thus, the algorithm takes into account the speed and direction of the approaching drone and calculates (and flies!) a velocity to maximize the distance between the two drones.

If move vertical is chosen as the avoidance action, the algorithm moves vertically regardless of the approaching drone's velocity. Whether to ascend or descend depends on the relative altitude of the two drones.

D. Power and timing technical details of RID transmitter

The broadcast power of the RID transmit modules is designed to meet or exceed the FAA regulations reflected in the ASTM specification [21] of +3 dBm. The minimum rate according to that same standard is also 1 Hz. We measured the rate and found it to be around 1 Hz.

The loop rate of the Ardupilot firmware is generally between 200-1000 Hz, depending on the version of the microcontroller used. Therefore, the collision avoidance decision is generally made in less than a few milliseconds. Since the RID rate is 1 Hz, that is the slowest part of the loop. Therefore, there may be up to 1 second delay from when the approaching zone is physically inside the detection perimeter, to when the flight controller makes a decision to avoid. After that, the kinetic response of the drone and the default speed parameters of the drone will determine the response time. The implications of this for the maximum speed of approaching drone that can be detected were already discussed above.

V. Flight Demonstrations

In order to demonstrate collision avoidance functionality, we have performed multiple flight demonstrations using two drones: The custom RID receiving drone (which we call the ”UCIRID” drone), and a DJI mini 3 or 4 Pro with a RID transmitter mounted on it (Holy Stone or Ruko brands), which we will call the ”DJI” drone. Both modes (DJI stationary, UCIRID drone flying toward it, and UCIRID drone stationary, and DJI flying towards it) were demonstrated with a variety of parameters for the hockey-puck ”bubble” size (10,30,50 m horizontal, 10 m vertical), and a variety of vehicle speeds, as well as four different pre programmed responses: warn, avoid by horizontal flight, avoid by vertical flight (climb/descend), and RTL.

In all flights, the RID drone log files were saved to disk, as well as the DJI drone log files. The RID drone low resolution video feed (720×480) and the DJI high-resolution 4K video were also saved to disk. All these files were analyzed post-flight, and the plots were generated as shown below. The raw video and log files are provided as supplementary information. A summary table of the flights is shown in Table II.

A. RID pilot prospective (FPV monitor)

The most immediate indication to the pilot of a pending collision is shown on the OSD on the FPV video. This is an immediate, low-cost, long-range (many km) method to give information and requires only the analog FPV receiver. Later we discuss the digital telemetry as an optional second mode of feedback to the pilot.

Shown in Fig. 5A is the FPV monitor view for one of the test flights. Note that the video quality is low since this is an analog feed. In fact, the approaching drone cannot be seen on the FPV monitor. This is the point. The UCIRID drone detects the approaching drone long before it is visible on the low-resolution feed. Before the DJI drone violates the collision bubble, the UCIRID drone pilot receives the xy and z distance to the approaching DJI drone. The perimeter was set at 30 m and the OSD could be shown to detect the approaching drone, even though the approaching drone was not visible on the monitor. Once the 30 m perimeter was breached, the drone took avoidance action (in this case, changing the mode to RTL), and placed a warning on the OSD, displaying ”TAKING FAIL ACTION” (Fig. 5B).

This demonstrates a hands-off, automated collision avoidance mode with zero pilot input during the flight and almost instantaneous pilot notification of the impending collision threat, and the automated anti-collision action. Thus, this demonstrates autonomous drone-drone collision avoidance for the first time in the air using broadcast RID.

B. RID pilot prospective (Ground control station through digital telemetry)

We have used dronebridge Wi-Fi based telemetry for a digital connection to the flight controller from the laptop running the ground control station, either Mission Planner or QGroundControl. In Mission Planner, if there is a collision detection or drone detected, the pilot is also notified.

The GCS is cluttered with information and the Wi-Fi range is only about 100 m. However, in contrast, the FPV signal

TABLE II
COLLISION AVOIDANCE MODE DEMONSTRATIONS

Collision avoid mode	**UCIRID drone**	**DJI drone**	**Hor. bubble radius (m)**	**Vertical bubble (m)**	**# of times demon-strated**
Warn only	moving	stationary	30	10	2
Warn only	stationary	moving	30	10	2
Move horizontal	moving	stationary	10	10	1
Move horizontal	stationary	moving	10	10	1
Move horizontal	moving	stationary	30	10	3
Move horizontal	stationary	moving	30	10	2
Move horizontal	stationary	moving	50	10	1
Move vertical	moving	stationary	50	10	1
Move vertical	stationary	moving	10	10	1
RTL	stationary	moving	30	10	1
				Total	**15**

Drone detected, 33 m away (not in avoidance perimeter)

Drone inside avoidance perimeter (collision avoidance action taken).

Fig. 5. Flight demonstration of collision avoidance. The upper panel shows the detected approaching drone distance prior to collision avoidance being triggered. The lower panel demonstrates avoidance action when the approaching drone broaches the perimeter (30 m in this case), causing the flight mode to be switched to RTL. Note due to the low inherent quality of the video, the DJI drone is not visible in the FPV feed, even though the UCIRID drone has already detected long before it is visible in the screen. This clearly demonstrates an advantage of broadcast RID as a DAA tool over visual see and avoid.

is one way, requires no negotiated connection (in contrast to Wi-Fi), is longer range (several km) and more robust.

In sum, we have presented two methods of notifying the UCIRID pilot with an imminent collision and with autonomous action to prevent an impending collision, and demonstrated both in flight demonstrations for the first time using broadcast RID.

In order to provide additional confirmation and verification of the system, we have extended our experimental campaign and data analysis beyond this initial proof of concept, described below.

C. Extended Flight demonstration 1: 10 m xyz horizontal avoid

In these flight demonstrations, the bubble was set to a radius of 10 m of the disk and a height of 10 m. The UCIRID drone flew horizontally to avoid collision when the bubble was breached by the DJI drone.

1) Extended Flight demonstration 1a: UCIRID drone flies towards DJI (in air, hovering): In this flight demonstration, the DJI drone hovered in the air (GPS loiter). The UCIRID drone was flown towards it, and when the distance was within the bubble, the UCIRID drone took avoidance action. The flight track from the log files of both drones is shown in Fig. 6.

The flight tracks of the two drones are shown as plots of the trajectories (x,y and z) of the two drones with respect to time in Fig. 7. This makes it possible to clearly see that the avoidance maneuver is initiated. Similar plots for the velocities of the two drones are shown as well. In this case, the DJI drone was stationary. The UCI drone flew towards it and when it breached the perimeter, it turned around and flew away from it. The altitudes did not change.

We next discuss more thoroughly and in detail information on when (around what time) the avoidance maneuver is really initiated. In Fig. 6 it is clear that the UCI drone was flying towards the (stationary) DJI drone when it went into collision avoidance mode (blue curve). It had to first slow down and then reverse course. For clarity, in Fig. 7, we have set T=0 as the time it detected a possible collision. The Ardupilot log file

This article has been accepted for publication in IEEE Aerospace and Electronic Systems Magazine. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MAES.2026.3668143

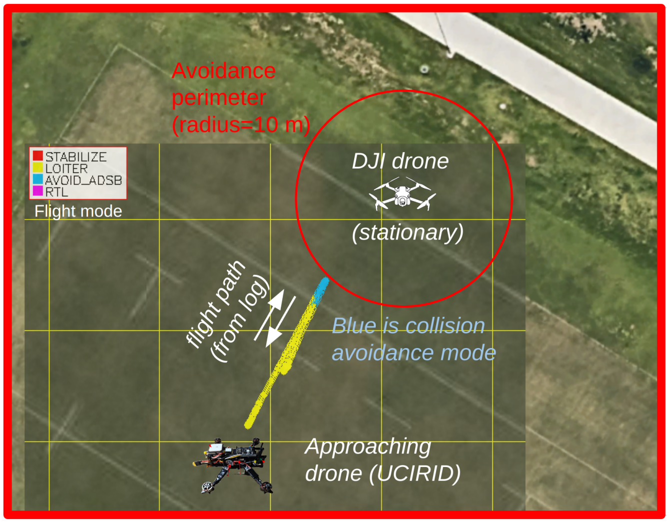

Fig. 6. Flight demonstration of collision avoidance. The DJI drone was in stationary mode, and the UCIRID drone flew towards it. Upon breach of the bubble, the UCIRID drone took autonomous collisions avoid action, flying away from the DJI drone.

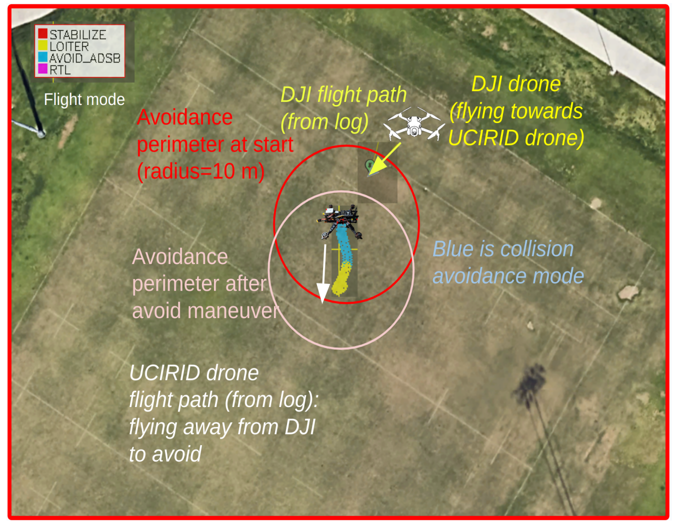

Fig. 8. Flight demonstration of collision avoidance. The UCIRID drone was in stationary mode, and the DJI drone flew towards it. Upon breach of the bubble, the UCIRID drone took autonomous collisions avoid action, flying away from the DJI drone.

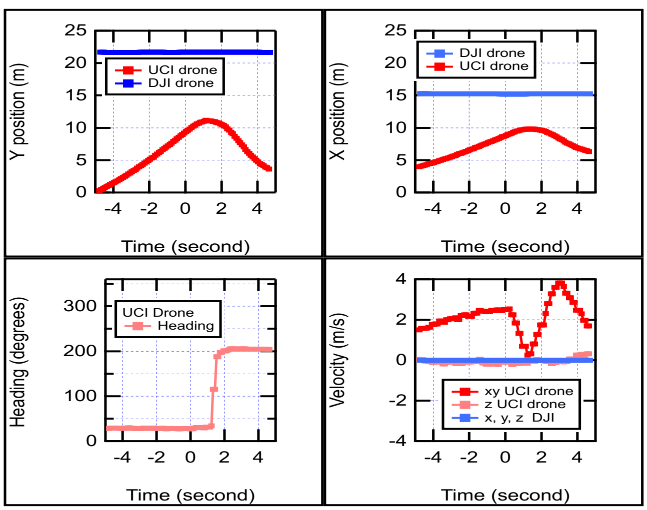

Fig. 7. Time plots of the position and velocity of the two drones during the first flight demonstration. The DJI drone did not move at all. The altitudes were the same.

provides data every 200 ms. In Fig. 7, the avoidance mode is entered (meaning the drone has been detected as a collision threat and initiated the avoidance maneuver) at T=0 s on the axis. Within one or two data points in the log (e.g. under 400 ms), the velocity started to slow down significantly. This is clear in the bottom right panel of Fig. 7. Finally after about 1 second, when it had slowed to a stop, it reversed course (bottom left panel) and then flew away from the (stationary) DJI drone. The UCI drone continued to fly away from the (stationery) DJI drone until the collision bubble was cleared.

2) Flight demonstrations 1b: DJI drone flies towards UCIRID (in air, hovering): In this flight demonstration, the UCIRID drone was hovering in the air (GPS loiter). The DJI drone flew towards it and when the distance was within the bubble, the UCIRID drone took avoidance action. The flight track from the log files of both drones is shown in Fig. 8. A high resolution 4k video at 3x zoom taken from the DJI is provided in the supplementary information.

The flight tracks of the two drones are shown as plots of the trajectories (x,y and z) of the two drones with respect to time in Fig. 9. Similar plots for the velocities of the two drones are shown as well. In this case, the DJI drone was flown towards the UCI drone. The UCI drone was stationary until the DJI drone breached the perimeter. At that point, the UCI drone flew away. The altitudes did not change.

We next discuss more thoroughly and in detail information on when (around what time) the avoidance maneuver is really initiated. In Fig. 9, we have set the T=0 point as exactly when the UCI drone went into collision avoidance flight mode, determined from the flight logs. It is clear from the bottom right panel that the velocity of the UCI drone began increasing within one or two data points (each data point corresponds to about 200 ms ellapsed time) of the time it detected a possible collision. The UCI drone continued to fly away from the (moving) DJI drone until the collision bubble was cleared.

Screenshots are shown in Fig. 10, showing the UCI drone flying away from the approaching DJI drone.

D. Extended Flight demonstrations 2,3: xy 30,50m z 10m hor avoid

As the 10m flight 1 mode, we demonstrated collision avoidance with a 30 m perimeter horizontal avoid mode for both cases (DJI stationary, or UCIRID stationary). This was demonstrated 5 times. A high resolution 4k video at 3x zoom taken from the DJI is provided in the supplementary information. Screenshots are shown in Fig. 11, showing the UCI drone flying away from the approaching DJI drone. The UCIRID drone is difficult to see in the pictures at this distance, emphasizing the advantage of RID broadcast as an alternative Detect and Avoid (DAA) method. We also performed similar

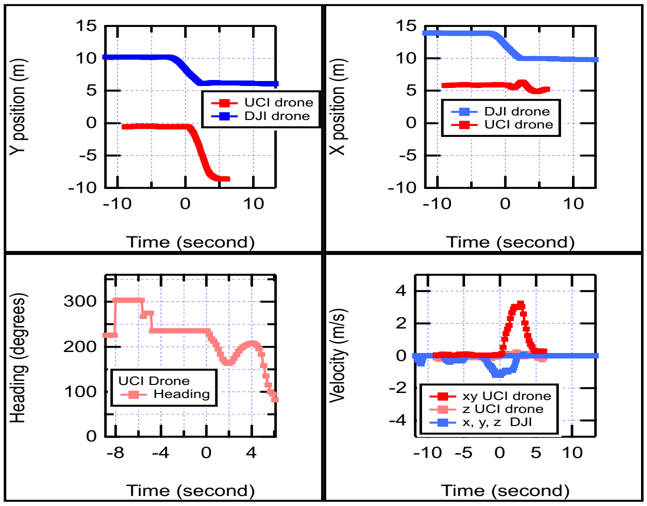

Fig. 9. Time plots of the position and velocity of the two drones during the first flight demonstration. The DJI drone flew towards the UCI drone, which initially was stationary. The UCI drone flew away when the DJI drone breached the perimeter. The altitudes were the same.

Fig. 10. Screenshots of video taken from DJI drone as it was flown towards UCIRID drone. The UCIRID takes collision avoidance action when the DJI drone breaches the predefined 10 m perimeter.

Fig. 11. Screenshots of video taken from DJI drone as it was flown towards UCIRID drone. The UCIRID takes collision avoidance action when the DJI drone breaches the predefined 30 m perimeter.

flight demonstrations with a 50 m bubble, but that was too far for the DJI camera to see.

E. Extended Flight demonstrations 4: xy 10m z 10m vertical avoid

1) Extended Flight demonstration 4a: UCIRID drone flies towards DJI (in air, hovering): In this flight demonstration, the DJI drone hovered in the air (GPS loiter). The UCIRID drone was flown towards it, and when the distance was within the bubble, the UCIRID drone took a vertical avoidance action. The flight track from the log files of both drones is shown in Fig. 12.

2) Extended Flight demonstration 4b: DJI drone flies towards UCIRID (in air, hovering): In this flight demonstration, the UCIRID drone was hovering in the air (GPS loiter). The DJI drone flew towards it and when the distance was within the bubble, the UCIRID drone took a vertical avoidance action.

This article has been accepted for publication in IEEE Aerospace and Electronic Systems Magazine. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MAES.2026.3668143

Fig. 12. Flight demonstration of collision avoidance. The UCIRID drone flies towards the stationary DJI drone. When the collision perimeter is breached, the UCIRID drone climbs to avoid collision.

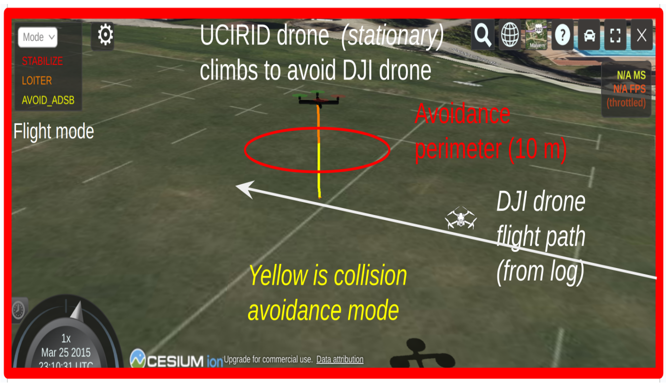

Fig. 13. Flight demonstration of collision avoidance. The UCIRID drone flies towards the stationary DJI drone. When the collision perimeter is breached, the UCIRID drone climbs to avoid collision.

The flight track from the log files of both drones is shown in Fig. 13.

F. Extended Flight demonstration: Range test (both drones in the air)

With both drones in the air, at our airfield we were only able to range test out to 200 m before exceeding the permissioned airspace. We found that the actual range was strongly dependent on the antenna orientation, the drone orientation, and the location of the antenna on the drone. With the RX antenna on the front of the drone and no obstructions, the UCIRID drone detected DJI up to 200 m away without problems. However, that dropped to about 100 m if the UCIRID drone was rotated in the air so that the drone was between the antenna and the DJI. The ASTM standard does not specify a range, only a power, and these results are consistent with what is to be expected for Bluetooth 4. The ASTM standard reports a range of 400 m for Bluetooth 4 in a rural environment, consistent with our results.

VI. Discussion

A. Mathematical point of view: System model definition, accuracy evaluation

From a mathematical point of view, our system model is simply 1) the UCIRID drone with a given xyz location (lat/lon/altitude) determined by the on-board GPS as well as any other pertinent information such as barometer, determined using an Extended Kalman Filter (EKF) algorithm state estimation theory built into Ardupilot at level AHRS2 (Attitude and Heading Reference System); and 2) the DJI drone with a given xyz location and velocity, as determined by RID transmit according to the ASTM RID standards In the RID standard, there is an "accuracy" estimate range broadcast for each packet, for both horizontal and vertical. We did not use that in our work. It would be an important improvement on our work to include that in collision avoidance decisions. That is left as a project for future research.

However, we can estimate the accuracy on the basis of known GPS receivers. The US commitment to GPS minimum performance [38] lists 8 m 95 percent confidence Horizontal Error and 13 m 95 confidence Vertical Error, although in practice the performance is usually better. For example, the BN880 used in this work has a 50 % Circular Area of Probability of 50

Finally, our system model consists simply of a perimeter avoid bubble. Since this is a proof-of-concept testbed, we did not refine the system model for collision estimate probability to change based on other real-time information. Again, that is left as a project for future research.

B. Limitations of this demo

Although we have provided proof-of-concept DAA with broadcast RID, there are some limitations of the particular implementation design decisions made in this case that could be improved on in future versions.

All of the code is on github with usage tutorials.

1) Bluetooth vs Wi-Fi: This proof of concept used only broadcasts transmitted over Bluetooth. RID requirements can be met by wireless or Bluetooth broadcast. An extension to Wi-Fi should be straightforward, since the ESP32 RID receiver on board also has built-in Wi-Fi.

2) Antenna: The development board had a PC antenna that was not optimally placed, with blockage for many drone orientations, leading to a reduced detection range. This could be improved by placing an antenna more strategically on the drone.

3) Number of drones: In this software, we are limited to avoiding one approaching drone. In principle, this could easily be extended to hundreds of drones, limited only by the on-board memory and processing capabilities of the microcontroller.

4) Technology Readiness Level (TRL) of the code: The code reuses the ADS-B collision avoidance code already in the "stock" ArduPilot release (v4.4). Although this demonstration has proved the concept, if it were to be merged into ArduPilot, the RID modules would need to be reconfigured as a completely separate package from the ADS-B modules. At the moment, the way the code base is configured, they are mutually exclusive. Furthermore, the ArduPilot parameters would need to be supplemented/redefined to include the RID parameters. All of these are possible, in principle, with additional effort and could lead to simultaneous ADS-B and RID

This article has been accepted for publication in IEEE Aerospace and Electronic Systems Magazine. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MAES.2026.3668143

avoidance in one firmware code base. The MAVLink protocol would not need to be revised, as it contains enough parameters to accommodate both.

C. Advantages/disadvantages of this approach in general

1) Non-RID (FRIA, sub 250): This DAA method only functions for drones using broadcast RID. In the USA, sub-250 g non-registered drones are not required to be equipped with RID, and FRIAs (federally recognized identification area) do not require RID.

2) Range: The hypothetical range of BT5 is up to 1 km and Wi-Fi and BT4 are typically 300 m. This limits the range of DAA using RID. Fast-moving approaching drones that would need to be detected farther out than around 300 m for safe avoidance would not be well served by this approach.

3) Interference: Interference is unlikely to be a problem, as Wi-Fi and Bluetooth protocols have built-in immunity to interference with methods such as channel hopping, time domain multiplexing, etc. In addition, the RID transmit rate is limited to 1 Hz to avoid channel capacity saturation.

D. Recent events

1) Drone - manned incidents: Several recent events demonstrate the possible importance of DAA with drones. The first was the collision of a DJI drone with a manned firefighting aircraft in the Los Angeles area during the fire season of early 2025 [39]. Technologically, it is possible to fit an ADSB receiver on a drone. In fact, Ardupilot already has that codebase, and it has been demonstrated elsewhere (refs). One of the DJI drone models (DJI Air 3) has an already integrated ADSB receiver and provides pilot notification. Therefore, it is possible that collision could have been avoided if an integrated ADSB receiver was on the drone and was used as automated DAA. However, for this to be widespread of use, it must be generalized to UAV-UAV and UAV-manned, and a broadcast cannot always be assumed.

2) Drone swarms: In Florida, during a drone light show [40], several drones collided with one another. A further mishap resulted in serious injury to a spectator. At present, drone light shows do not have UAV-UAV collision avoidance. Although implementing this work in a drone light show or other drone swarm uses would be more challenging, it could enable UAV-UAV anticollision.

3) Summary: In summary, this paper demonstrates a strategy for collision avoidance and DAA with broadcast RID, but it must be viewed as one tool in the toolbox for an integrated collision avoidance strategy that will be required for the full integration of manned and unmanned aircraft into national and international airspace.

E. Future research: Towards a comprehensive collision avoidance system

This work represents a proof of concept demonstration in hardware and software of a straightforward collision avoidance algorithm. Future research can address different avoidance algorithms and use our approach to easily implement them. Multiple scenarios exist, such as the fixed wing vs. quadrotor of both the threat and threatened drone, different approach velocities, different ranges for RID systems, different desired avoidance parameters (such as min distance between drones), and different time responses (slow vs. fast). An exhaustive study of all possible avoidance scenarios is left to future studies. However, all future studies can use the approach in this paper as an initial pioneering proof of concept.

However, this is the first such study to include DAA in an existing, widely used, open source code base (Ardupilot). Since Ardupilot is installed on more than a million vehicles, this study has a very wide applicability to future potential implementations of this algorithm.

VII. Vision for use case

At this point, based on our experience with this technology in the field and in the trenches, we provide our recommendation for practical use cases. This is an opinion only, and so should only be considered as an editorial discussion. The biggest disadvantage of broadcast RID is the narrow window between the detection range (a few hundred meters) and the anticipated collision avoidance perimeter size (a few tens of meters). This is not a large margin for collision avoidance. Also, the range is not part of the ASTM RID specification, so it may be significantly less than a few hundred meters in practice. Therefore, we do not recommend it in general as the primary DAA technology for the general use case. Drone light shows and drone swarms may be an example where it is appropriate, since the relative speed between two drones is low and the range is short.

On the other hand, we have demonstrated the implementation of broadcast RID DAA with a very low-cost transceiver, under ten US dollars, that weighs only a few grams, with no additional hardware. The code is open source, i.e. free. So per drone the cost is nominal. Moreover, most drones these days have some form of radio communications hardware in the Wi-Fi/BT band for other purposes, such as remote control, video, telemetry, etc. It should be possible to add broadcast RID based DAA as a task without degradation to the other tasks, especially since most drones have powerful microprocessors on board for flight control. In this case, the extra cost per drone on the hardware and software side would be zero dollars. Thus, although we do not recommend this as the prime avoidance technology in most use cases, our work can provide the basis for a zero-cost, additional level of safety for drones as a secondary collision avoidance technology. This could be a very important backup system in case the envisioned network drone RID fails, a very likely scenario that must be taken into account in any future envisioned centrally managed integrated drone-manned aviation air traffic control system.

VIII. Conclusion

We have demonstrated an open source detect and avoid (DAA) system for unmanned aerial vehicles (UAVs) using broadcast Remote ID (RID) based on standard components. This system does not rely on any additional hardware other than one ESP32 receive board, which costs around ten US dollars and weighs only a few grams. There is no need for cloud connectivity, nor any additional on-board computer system. Since it is a lightweight (one hertz update rate) design, the extra code can easily be handled by the STM32 microcontroller on virtually any modern flight controller. This provides another tool in the DAA toolbox needed for safe integration of drones into the airspace system.

ACKNOWLEDGMENT

This project is supported by ASTM International through their University Design Project Grants and by UCI through a UROP (Undergraduate Opportunities Research Program) grant. The authors thank Shawn Davis of UCI EECS for technical assistance, as well as members of the AMA local club UAVS@UCI for assistance in spotting and flights.

REFERENCES

[1] P. J. Burke, "A Safe, Open Source, 4G Connected Self-Flying Plane With 1 Hour Flight Time and All Up Weight (AUW) $<$ 300 g: Towards a New Class of Internet Enabled UAVs," *IEEE Access*, vol. 7, pp. 67 833–67 855, 2019.

[2] Federal Aviation Administration and Transportation Security Administration, "Normalizing Unmanned Aircraft Systems Beyond Visual Line of Sight Operations: A Proposed Rule," Federal Register, Department of Transportation, Department of Homeland Security, Docket No. FAA-2025-1908; Notice No. 25-07, RIN 2120-AL82, RIN 1652-AA80, August 2025, 14 CFR Parts 36, 43, 45, 48, 89, 91, 107, 108, 119, 133, 135, 137, and 146; 49 CFR Parts 1540 and 1544.

[3] M. P. Owen, A. Panken, R. Moss, L. Alvarez, and C. Leeper, "ACAS Xu: Integrated Collision Avoidance and Detect and Avoid Capability for UAS," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019, pp. 1–10.

[4] L. E. Alvarez, I. Jessen, M. P. Owen, J. Silbermann, and P. Wood, "ACAS sXu: Robust Decentralized Detect and Avoid for Small Unmanned Aircraft Systems," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019, pp. 1–9.

[5] C. Muñoz, A. Narkawicz, G. Hagen, J. Upchurch, A. Dutle, M. Consiglio, and J. Chamberlain, "DAIDALUS: Detect and Avoid Alerting Logic for Unmanned Systems," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, 2015, pp. 5A1–1–5A1–12.

[6] M. Consiglio, J. Chamberlain, C. Muñoz, and K. Hoffler, "Concepts of Integration for UAS Operations in the NAS," Tech. Rep., 2012.

[7] J. T. Davies and M. G. Wu, "Comparative Analysis of ACAS-Xu and DAIDALUS Detect-and-Avoid Systems," Tech. Rep., 2018.

[8] S. Bijjahalli, A. Gardi, S. Hilton, and R. Sabatini, "A Novel Detect-and-Avoid Approach for UAS in Urban Environments," in *32nd Congress of the International Council of the Aeronautical Sciences, ICAS 2021*. International Council of the Aeronautical Sciences, 2021.

[9] L. R. Sahawneh, M. O. Duffield, R. W. Beard, and T. W. McLain, "Detect and Avoid for Small Unmanned Aircraft Systems Using ADS-B," *Air Traffic Control Quarterly*, vol. 23, no. 2-3, pp. 203–240, 2015.

[10] W. Khawaja, M. Ezuma, V. Semkin, F. Erden, O. Ozdemir, and I. Guvenc, "A Survey on Detection, Tracking, and Classification of Aerial Threats using Radars and Communications Systems," *arXiv:2211.10038*, 2022.

[11] M. ki, J. cha, and H. Lyu, "Detect and avoid system based on multi sensor fusion for UAV," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, 2018, pp. 1107–1109.

[12] M. U. de Haag, C. G. Bartone, and M. S. Braasch, "Flight-Test Evaluation of Small Form-Factor LiDAR and Radar Sensors for sUAS Detect-and-Avoid Applications," in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016, pp. 1–11.

[13] J. Riordan, M. Manduhu, J. Black, A. Dow, G. Dooly, and S. Matalonga, "LiDAR Simulation for Performance Evaluation of UAS Detect and Avoid," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 1355–1363.

[14] T. L. Molloy, J. J. Ford, and L. Mejias, "Detection of aircraft below the horizon for vision-based detect and avoid in unmanned aircraft systems," *Journal of Field Robotics*, vol. 34, no. 7, pp. 1378–1391, 2017.

[15] R. Sridhar, M. Murphy, and A. Lamprecht, "Towards Robust Certification of Computer-Vision-Based Detect and Avoid in UAS," in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, 2020, pp. 1–5.

[16] "The practical collision warning system." [Online]. Available: https://www.flarm.com/en/

[17] L. M. Schalk and N. Peinecke, *Detect and Avoid for Unmanned Aircraft in Very Low Level Airspace*. Cham: Springer International Publishing, 2022, pp. 333–351. [Online]. Available: https://doi.org/10.1007/978-3-030-83144-8_14

[18] M. Marques, A. Brum, S. Antunes, and J. G. Mota, "Sense and avoid implementation in a small unmanned aerial vehicle," in *2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO)*. IEEE, 2018, pp. 395–400.

[19] N. Ruseno, C.-Y. Lin, and W.-L. Guan, "Flight test analysis of utm conflict detection based on a network remote id using a random forest algorithm," *Drones*, vol. 7, no. 7, p. 436, 2023.

[20] "Remote Identification of Drones — Federal Aviation Administration." [Online]. Available: https://www.faa.gov/uas/getting_started/remote_id

[21] *ASTM F3411-19: Standard Specification for Remote ID and Tracking*, ASTM International Std., 2019, available: https://www.astm.org.

[22] E. Vinogradov and S. Pollin, "Reducing safe uav separation distances with u2u communication and new remote id formats," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 1425–1430.

[23] E. Vinogradov, A. S. B. Kumar, F. Minucci, S. Pollin, and E. Natalizio, "Remote id for separation provision and multi-agent navigation," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*. IEEE, 2023, pp. 1–10.

[24] A. Attin, S. Bonnedahl, Z. Wang, M. Wzorek, A. Lemetti, and A. Gurtov, "Secure remote id and detect-and-avoid in unmanned aerial systems: Modeling the maximum safe speed," in *2024 IEEE International Conference on Aerospace and Signal Processing (INCAS)*. IEEE, 2024, pp. 1–6.

[25] "DroneBridge/ESP32: DroneBridge for ESP32. A transparent short range wifi based telemetry (serial to WiFi) link. Support for MAVLink, MSP, LTM (iNAV) or any other protocol," *GitHub*. [Online]. Available: https://github.com/DroneBridge/ESP32

[26] "uci-overRID/RID: Remote ID receiver on ESP32," *GitHub*. [Online]. Available: https://github.com/uci-overRID/RID

[27] "uci-overRID/ardupilot: ArduPlane, ArduCopter, ArduRover, ArduSub source," *GitHub*. [Online]. Available: https://github.com/uci-overRID/ardupilot

[28] "Telemetry Logs — Mission Planner documentation." [Online]. Available: https://ardupilot.org/planner/docs/mission-planner-telemetry-logs.html

[29] A. Weinert, S. Campbell, A. Vela, D. Schuldt, and J. Kurucar, "Well-clear recommendation for small unmanned aircraft systems based on unmitigated collision risk," *Journal of air transportation*, vol. 26, no. 3, pp. 113–122, 2018.

[30] E. T. Lester and A. Weinert, "Three quantitative means to remain well clear for small uas in the terminal area," in *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE, 2019, pp. 1–17.

[31] N. Underhill and A. Weinert, "Applicability and surrogacy of uncorrelated airspace encounter models at low altitudes," *Journal of Air Transportation*, vol. 29, no. 3, pp. 137–141, 2021.

[32] M. W. Edwards and J. Mackay, "Determining required surveillance performance for unmanned aircraft sense and avoid," in *17th AIAA Aviation Technology, Integration, and Operations Conference*, 2017, p. 4385.

[33] A. D. Evans, M. Egorov, A. Anand, S. E. Campbell, S. Zanlongo, T. Young, and N. Sarfaraz, "Safety assessment of utm strategic deconfliction," in *AIAA Scitech 2023 Forum*, 2023, p. 0965.

[34] R. Weibel and R. J. Hansman, "Safety considerations for operation of different classes of uavs in the nas," in *Aiaa 4th aviation technology, integration and operations (atio) forum*, 2004, p. 6244.

[35] S. P. Cook, D. Brooks, R. Cole, D. Hackenberg, and V. Raska, "Defining well clear for unmanned aircraft systems," in *AIAA Infotech@ Aerospace*, 2015, p. 0481.

[36] A. Weinert, L. Alvarez, M. Owen, and B. Zintak, "Near midair collision analog for drones based on unmitigated collision risk," *Journal of Air Transportation*, vol. 30, no. 2, pp. 37–48, 2022.

[37] M. Omeri, R. Isufaj, and R. M. Ortiz, "Quantifying well clear for autonomous small uas," *IEEE Access*, vol. 10, pp. 68 365–68 383, 2022.

[38] *Global Positioning System Standard Positioning Service Performance Standard*, 5th ed., U.S. Department of Defense,

2020. [Online]. Available: https://www.gps.gov/technical/ps/2020-SPS-performance-standard.pdf

[39] A. Press, "Faa says drone collided with firefighting aircraft flying over palisades fire," *AP News*, 2025. [Online]. Available: https://www.usatoday.com/story/news/nation/2025/01/10/drone-firefighting-aircraft-palisades-fire/77587949007/

[40] National Transportation Safety Board, "Preliminary report: Investigation of uvify ifo small swarm light show drone orlando, florida," National Transportation Safety Board, Tech. Rep. NTSB Identification DCA25LA065, January 2024, accessed on 2025-01-15. [Online]. Available: https://data.ntsb.gov/carol-repgen/api/Aviation/ReportMain/GenerateNewestReport/199458/pdf