Aashay S Shah[1], Wentao Chen[1], Jason Ho[1], Samuel Jawhar[1], Tiantian Lu[1], Mengjie Xie[1], Pengfei Yan[1], and Peter Burke[1]

[1]Affiliation not available

February 12, 2025

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 takes avoidance action if a user-defined perimeter is breached (in 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.**

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.

Several techniques have been proposed for collision avoidance. For example, ADS-B Out of manned aircraft could be used by drones' on-board receivers for drone-manned aircraft avoidance. Networked drones could be managed by a cloud system for a centralized drone-drone collision avoidance model. Eventually, all flying vehicles, whether manned or unmanned, could be managed in an integrated system. The progress towards this goal is reviewed at the end of this paper in the prior art section.

One disadvantage of a cloud-based system is over-reliance on a single point of failure (the link from drone to cloud), as well as the possibility of non-cloud-connected drones in the air. In this paper, we demonstrate a "local" avoidance paradigm that is immune to cloud-based downtime and only relies on the local broadcast of one drone to another to avoid collision: drone-to-drone communication via broadcast Remote ID.

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 May 28, 2024, revised February 2025.

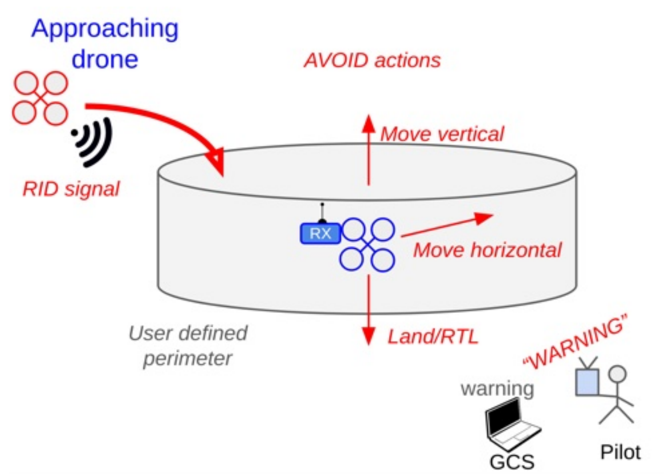


Fig. 1. Concept of operation (CONOPS). An RID receiver is integrated into the center drone, which detects any nearby drones within RID range. If the user-defined perimeter ("bubble" or "hockey puck") is breached, the drone takes avoidance action. The remote pilot is alerted to any drones in range on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

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 broadcast identification and location information while in flight under enforcement of the Federal Aviation Administration (FAA) [2]. Europe and Asia have similar requirements. Data such as drone real-time location/altitude, drone serial number, operator ID/location, etc. are broadcast via either Wi-Fi or Bluetooth from the drone. Therefore, our broadcast RID DAA paradigm could be widely deployed today with existing technology and within the current regulation framework. We demonstrate a proof-of-concept of this DAA paradigm in this paper 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.

II. Concept of operation (CONOPS)

Fig. 1 shows the concept of operation. The receiving drone has an RID receiver which receives all broadcast signals in range. The drone takes avoidance action if a user-defined perimeter is breached (in xy position 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.

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

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

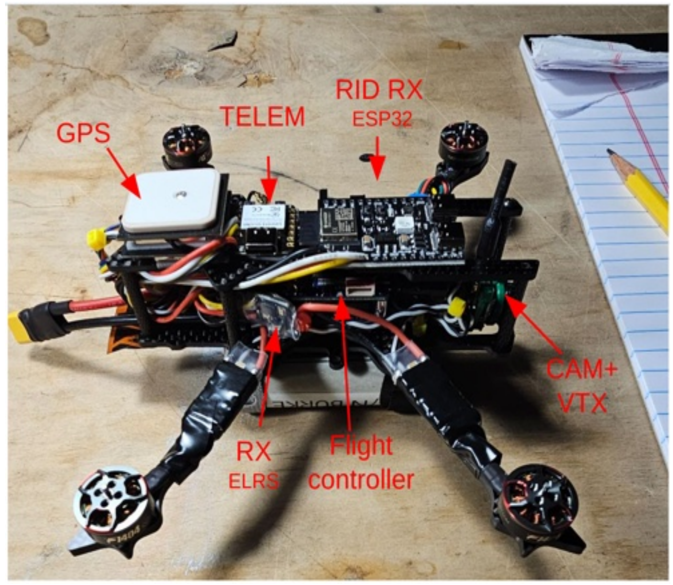


Fig. 2. Fully assembled drone used in this work. The RID receiver is shown mounted on the top of the drone.

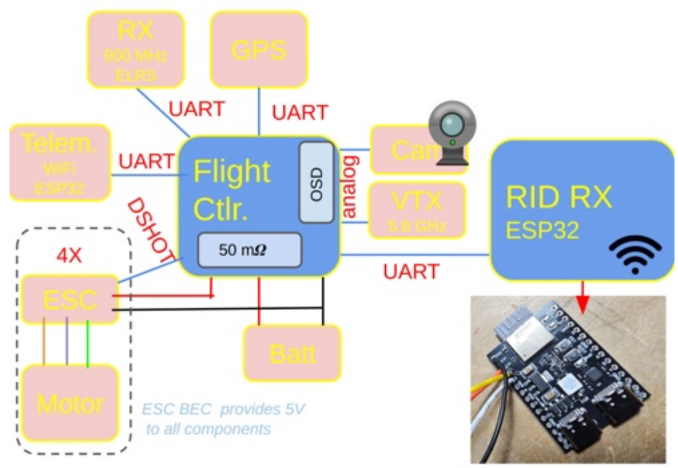


Fig. 3. Avionics. The RID receiver listens for broadcast signals from all drones in range, and sends the data to the flight controller over UART using the MAVLink protocol. A picture of the ESP32 board used is shown, with the Bluetooth PCB antenna clearly visible at top left.

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 also part of a widely aclaimed 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 [3], 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 software design is discussed next.

1) RID receiver: For the RID receiver, custom firmware was developed [4]. 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 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 [5]. 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 xy and z distance to the nearest approaching drone onto the on-screen display (OSD) through the video transmitter to alert the pilot and a warning is displayed on ground control stations (GCS).

C. Flight logs and post flight analysis

1) Flight logs of UCIRID drone: For the UCIRID drone (the drone with the 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 the flight, the GCS is also in constant communication with the drone using MAVLink telemetry, and it recorded log files locally on the PC in *tlog* [6] format. Finally, the OSD and video feed were recorded to an SD card on the video receiver for further post-flight analysis. Mission Planner software was used for post-flight analysis of the log files.

It should be noted that the analog first person view (FPV) cameras technology standard used has limited resolution. It is based on NTSC and PAL. PAL has a resolution of 720×576 at 25fps (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.

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 onboard log file, which can be uploaded to a commercial cloud log file analysis provider (Airdata UAV, www.airdata.com) and parsed via a web browswer. Because of 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 4k high quality 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. In our implementation of the software, the pilot can select the perimeter size on a flight-by-flight basis. We are exploiting code already developed for these algorithms in Ardupilot avoid ADSB code. We re-purpose code snippets for RID based DAA here. Once the perimeter is breached, based on the velocity of the approaching drone, a decision is made, discussed below. This is appropriate 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 broadcast RID standard, similar to the ADSB standard, requires 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.

The user defined input parameters (which can be set in Mission Planner and uploaded to the drone memory after the code is compiled on a flight by flight basis) are xy, z (Fig. 1), and Action. Possible actions are land, RTL (return to launch), move vertical, move horizontal. The first three occur as soon as the perimeter is breached. The third (move horizontal) depends the relative velocity of the two drones, and are discussed next. In the future, more sophisticated algorithms could be possible to implement with this system.

A. Move horizontally

If this 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 make the drones further apart. On the other hand, if the approaching drone speed is larger than 1 m/s, then the algorithm that is used is shown in Fig. 4. 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.

B. Move vertically

Here 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.

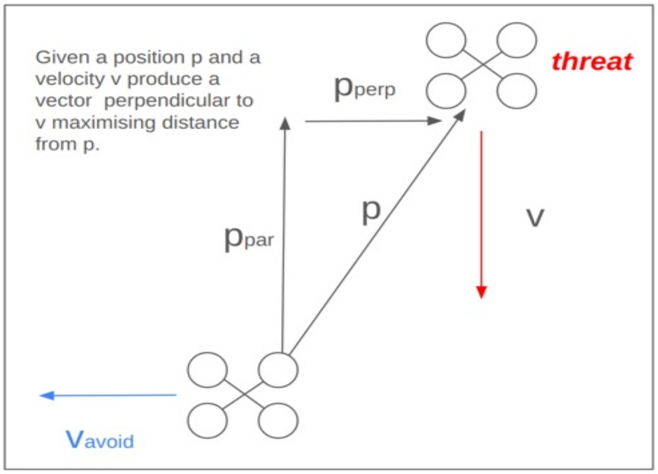


Fig. 4. Collision avoidance geometry.

C. Resolution

There is one more user parameter, which is the action/behavior of the drone after the approaching drone is no longer inside the perimeter. The user can select loiter (stay in place) or continue on the flight mode it was in prior to the collision, or land, or RTL. In our demos we select the continue option.

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 [7] 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 made generally within 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 a 1 second delay from when the approaching zone is inside the detection perimeter physically, to when the flight controller makes a decision to avoid. After that, the kinetic response of the drone as well as the default speed parameters of the drone will determined 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 towards it, and UCIRID drone stationary, and DJI flyings 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.

Collision avoid mode	UCIRID drone	DJI drone	Hor. bubble radius (m)	Vertical bubble (m)	# of times demonstrated
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

TABLE II
Table of flight demonstrations.

In all flights, the log files of the RID drone were saved to disk, as well as log files of the DJI drone. The RID drone low resolution video feed (720×480) and the DJI high resolution 4K video was also saved to disk. All these files were analyzed post flight and the plots were generated 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. Prior to the DJI drone violating the collision bubble, the UCIRID drone pilot is given 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 in the monitor. Once the perimeter of 30 m 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 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.

In a second flight demonstration, the drone was configured to move horizontally to avoid an oncoming detected drone. Fig. 6 shows the flight path and perimeter of the drone, recorded in the log files, plotted on a map overlay. When the approaching drone was outside the perimeter, only a warning was given. Upon breaching the perimeter, the drone moved

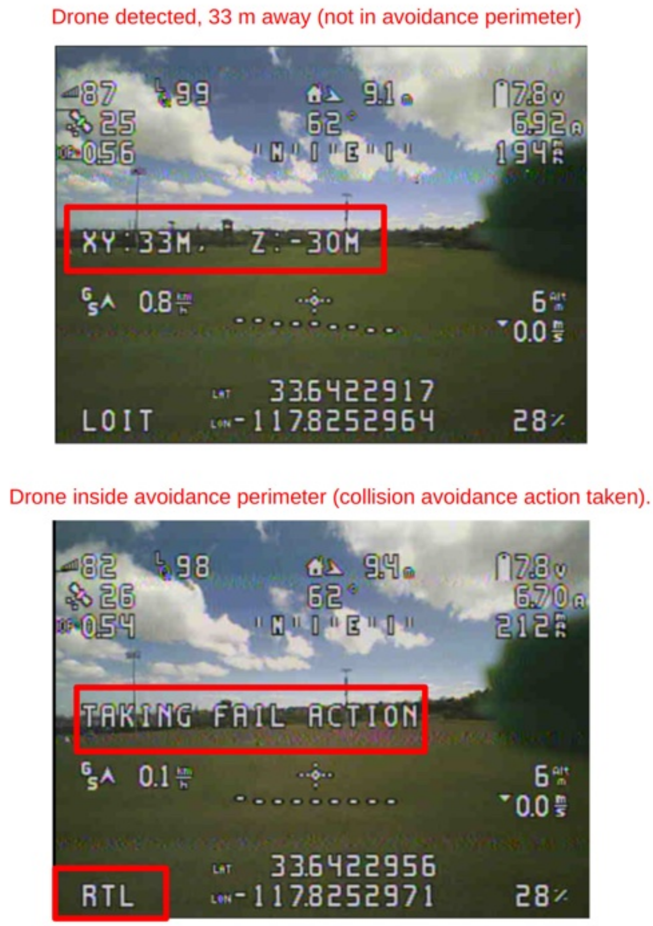


Fig. 5. Flight demonstration of collision avoidance. The upper panel shows the detected approaching drone distance prior to collision avoidance being triggered. The lower panel demonstrates avoidance action when the approaching drone broaches the perimeter (30 m in this case), causing the flight mode 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.

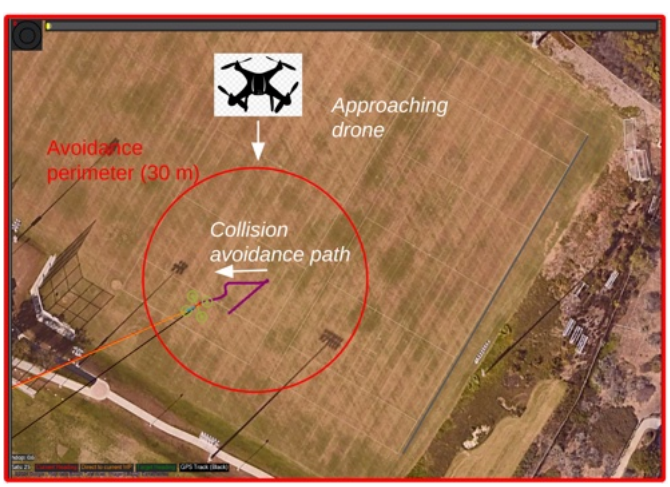


Fig. 6. Flight demonstration of collision avoidance. The purple path is the drone's flight path. The initial straight flight to the northeast was a positioning path. The drone hovered at the tip until the collision avoidance criteria was met, and flew west to avoid the approaching drone.

perpendicular to the approaching drone's trajectory, as shown in the collision avoidance path in the figure.

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 range of Wi-Fi is only about 100 m. However, in contrast the FPV signal 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 notifitying the UCIRID pilot with and impending collision and with autonomust 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 10 m radius of the disk and 10 m height. 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 was hovering 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. 7.

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 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. 8. A high resolution 4k video at 3x zoom taken from the DJI is provided in the supplementary information. Screenshots are shown in Fig. 9, 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 collisaion 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. 10, showing the UCI drone flying away from the approaching DJI drone. The

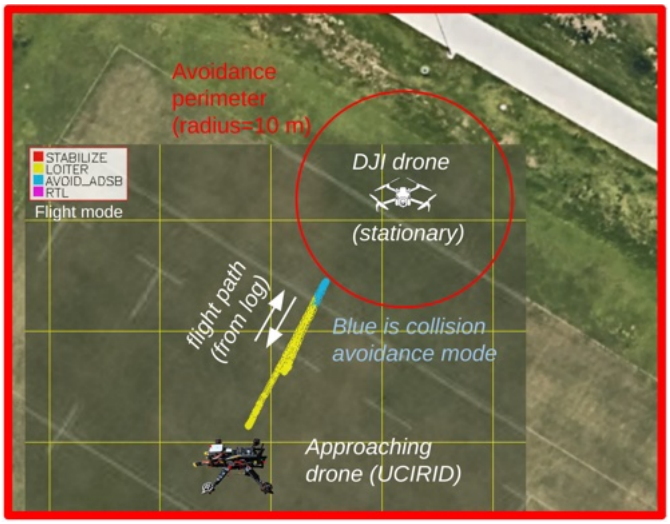


Fig. 7. Flight demonstration of collision avoidance. The DJI drone was in stationary mode, and the UCIRID drone flew towards it. Upon breach of the bubble, the UCIRID drone took autonomous collisions avoid action, flying away from the DJI drone.

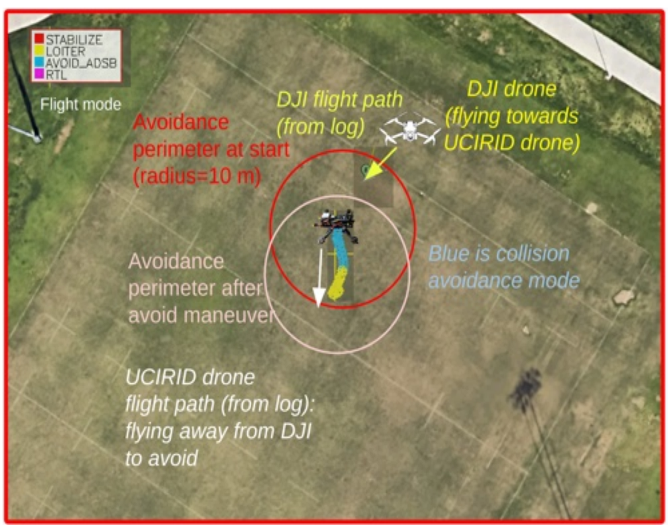


Fig. 8. Flight demonstration of collision avoidance. The UCIRID drone was in stationary mode, and the DJI drone flew towards it. Upon breach of the bubble, the UCIRID drone took autonomous collisions avoid action, flying away from the DJI drone.

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 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 was hovering in the air (gps loiter). The UCIRID drone was flown towards it, and when the distance was within the bubble, the UCIRID drone took vertical avoidance action.

Fig. 9. 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.

The flight track from the log files of both drones is shown in Fig. 11.

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 was flown towards it, and when the distance was within the bubble, the UCIRID drone took vertical avoidance action. The flight track from the log files of both drones is shown in Fig. 12.

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 the actual range depended strongly 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 the DJI up to 200 m away with no 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 400 m range for Bluetooth 4 in a rural environment, consistent with our results.

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 30 m perimeter.

Fig. 11. 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. 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.

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 xzy 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 tweek or improvement on our work to include that in collision avoidance decisions. That is left as a project for future research.

We can however estimate the accuracy, based on known GPS receivers. The U.S. commitment to GPS minimum performance [8] 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 CEP of 2 m horizontal.

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

B. Towards more sophisticated collision avoidance algorithms

The algorithm demonstrated in this paper is a simple perimeter breach algorithm. This is appropriate for slow-moving approaching drones. However, this is only a proof of concept. With the technology demonstrated in this paper, in principle, any algorithm could be implemented with simple code change.

For example, in ArduPilot’s algorithm for ADS-B-based avoidance with manned aircraft, the algorithm predicts aircraft position based on current velocity and distance, and proactively (based on predicted future trajectories) takes avoidance action even if the manned aircraft is not in imminent danger of collision. This is more appropriate for fast-moving aerial vehicles (either manned or unmanned), but less so for slower-moving quad-rotors.

In future work, we plan to explore additional anti-collision algorithms and tailor the algorithms to the particular nature of the specific approaching vehicle.

C. 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 either Wi-Fi or Bluetooth broadcast. An extension to Wi-Fi should be straightforward, since the on-board ESP32 RID receiver 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 reduced detection range. This could be improved on 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). While 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 parameters of ArduPilot would need to be supplemented/redefined to include RID parameters. All of these are possible in principle with additional effort, and could lead to simultaneous ADS-B and RID avoidance in one firmware code base. The MAVLink protocol would not need to be revised, as it contains enough parameters to accommodate both.

D. 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 typically 300 m. This limits the range of DAA using RID. Fast-moving approaching drones that would need to be detected further 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. Also the RID transmit rate is limited to 1 Hz to avoid channel capacity saturation.

E. 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 fire-fighting manned aircraft in Los Angeles area during the fire season of early 2025 [9]. Technologically, it is possible to fit an ADSB recevier onto 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 ADSB receiver already integrated and gives 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 [10], several drones collided with one another. A further mishap resulted in severe injury to a spectator. At present, drone light shows do not have UAV-UAV collision avoidance. While implementing this work in a drone light show or other drone swarm uses would be more challenging, it could enable UAV-UAV anti-collision.

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 full integration of manned and unmanned aircraft into the national and international airspace.

F. Future research: Towards a comprehensive collision avoidance system

This is not a paper about all possible collision avoidance algorithms, or even the best one. It is a simple proof of concept demonstration in hardware and software of a simple one. Future research can address different avoidance altorithms, and use our approach to easily implement them.

Furthermore, this is not a paper about all possible collision avoidance scenarios. Multiple scenarios exist, such as 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 study. However all future studies can use the approach in this paper as the initial, pioneering proof of concept.

Futhremore, this is the first such study to impplement the DAA in an existint, widely used, open source code base (Ardupilot). Since Ardupilot is installed on over a milllions vechicles, this study has a very wide applicabiliyt in future potential impelmentatinons of this algorithm.

G. Comparison to prior art

The idea of RID for DAA is not new. In fact it was listed in the ASTM RID standard [7] as a potential use case: ”A UAS operating under BVLOS would like to avoid a path conflict with another UAS.” Below we discuss prior art in this general direction.

Owen et al. [11] and Alvarez et al. [12] built upon FAA supported work on the ACAS X collision avoidance system for manned aircraft to bring such detect and avoid capabilities to unmanned aircraft systems (UAS). Muñoz et al. [13] developed a reference implementation of the functional specifications outlined in NASA's concepts for the integration of UAS into the U.S. National Airspace System [14]. Davies and Wu [15] performed a comparative analysis between the UAS DAA implementations by Owen et al. [11] and Muñoz et al. [13], and found comparable DAA performance, but with some notable discrepancies in suggestive guidance. All such DAA systems utilized surveillance data from sources such as ADS-B and radar.

Ki et al. [16] demonstrated a multi-sensor DAA system that performed depth estimation utilizing 2D lidar sensing for initial obstacle detection and a stereo camera for depth map analysis, and de Haag et al. [17] evaluated the in-flight performance of small form factor lidar and radar sensors for potential use in small UAS applications, noting robustness in multi-sensor solutions in more mature DAA technologies for medium to large UAS. Riordan et al. [18] proposed digital simulation for evaluation of the performance of lidar configurations in UAS DAA applications, confirming challenges in lidar detection of small uncooperative objects due to inhomogeneous point cloud generation. Molloy et al. [19] developed a fully camera vision-based system to detect uncooperative aircraft, including those appearing below the horizon, and Sridhar et al. [20] evaluated a number of vision-based object detectors for use in detecting aircraft in a DAA setting for UAS.

Bijjahalli et al. [21] proposed an analytical UAS DAA approach accounting for uncertainty in performance of communication, navigation, and surveillance systems associated with aircraft involved in a DAA scenario as well as potential wind and weather conditions, and demonstrated performance in simulated UAS scenarios involving radar and ADS-B. Sahawneh et al. [22] implemented a DAA approach for UAS based entirely on ADS-B and validated its potential in Monte Carlo simulations. They further propose that with the deployment of such ADS-B-based DAA systems, the mandating of ADS-B capabilities for all aircraft in the United States would facilitate a safe integration of UAS into the U.S. National Airspace System. Schalk and Peinecke [23] and others [24] also propose the use of FLARM [25] as an alternative to ADS-B for use in DAA systems for very low level (VLL) airspace in the European Union.

Khawaja et al. [26] surveyed the landscape of UAS detection systems relying primarily on radar, as well as systems utilizing joint communication-radar (JCR) and systems that passively monitor the communication transmissions to/from UAS as well as the reflections and interferences in existing external communication signals resulting from the presence of UAS. They discuss popular such communication systems including GSM/LTE/5G, satellite, Wi-Fi, Digital and Analog Audio Broadcast, Digital Video Broadcast, and RF.

The paper [27] demonstrates Traffic Management for unmanned aerial systems using Network Remote ID, a system which is currently not required by the FAA. Network Remote ID requires the drone to have an always-on internet connection. This system transmits the UAV data to a Traffic Management System to warn pilots of potential collisions. Our system on the other hand utilizes Broadcast Remote ID which simply transmits UAV information over Wi-Fi and Bluetooth systems and is currently mandated by the FAA. The demonstrated system in the paper provides warnings to pilots about potential collisions and relies on them for action to be taken while our system takes action in an automated manner.

Refs. [28], [29] do an outstanding job of simulating the correct ”bubble size” for RID based DAA based on a variety factors such as wireless range, airframe size and speed, and position accuracy. The call it the ”uNMAC: UAV Near Mid-Air Collision” volume, and base it on additional factors not present in the existing RID standard, such as aircraft size, localization precision UAV speed/velocity, and the capabilities of wireless technologies. They use simulations to propose additional (future) RID information in the broadcast. For example, they propose 10 Hz rather than 1 Hz update rate in future RID revised standards. Our work is related to those two papers as follows: Our work is the first system, software, hardware, and flight demonstration of broadcast RID DAA, based on existing (not future) RID standards, for a specific ”bubble” size. (Note in this work, we make no attempt or claim to optimize the ”bubble size”.) Also, in contrast to Ref. [28], [29] , our work does propose and implement a collision avoidance technique, albeit a simple proof of concept one. To quote from Ref. [28], [29] ”This study does not implement a collision avoidance technique...” Ref. [28], [29] work uses simulations to fine tune the bubble size (which could be done in our system as a simple software extension), as well as proposes future changes to RID broadcast standards for improved broadcast RID DAA. So in sum, ref. [28], [29] is theoretical simulation of hypothetical bubble geometries and sizes, whereas this work is real world actual demonstration of specific proof of concept example bubble geometries and sizes. Both approaches are complementary and necessary for future implementation of broadcast RID for DAA in real world systems. In particular, future research needs to take our hardware/software real world demonstration and test all of the parameters of ref [28], [29]. Such a work is beyond the scope of this work, which is proof of concept demonstration with open source instructions for anyone in the drone community to use to test the simulations of ref [28], [29] as well as any other proposed collision avoidance algorithms or shapes. Thus, our work presents a very powerful testbed platform which will benefit the entire DAA RID research community, enabling them to test, and fine tune the most appropriate collision avoidance algorithms for all current and future drone technology.

Ref [30], which was published during the review of this paper, simulates and models the maximum safe flying speed for a given collision avoidance algorithm, something we discussed above in the update rate section. Therefore, research is still ongoing on the relationship between the update rate of RID (currently 1 Hz) and the speed of approaching UAVs to avoid. A related real-world indoor (non-GPS) demonstration of a related collision avoidance system was also published during the review of this paper in [31]. Note also ref. [30] clearly states "Future research should focus on real-world testing", which is exactly what this paper demonstrates for the first time.

Finally, the actual hockey puck/bubble size that should be used for well-clear has been quantified for manned-UAV collision avoidance by the extensive work of Weinert and colleagues [32]–[38]. He found 2000 feet horizontal and 500 feet vertical was the appropriate hockey puck for well clear. However, no such connsensus 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 [39], 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 then MAC bubbles. For example, in Ref. [40] (Tables 4,5) recommend 60-85 m as the bubble size for well-clear drone-drone collisions. Therefore, our work is a demonstration for a specific hockey puck size that is right in line with that found by Weinert as an initial estimate. Much more research is still needed on the appropriate hockey puck size for drone-drone collision avoidance. Ours is the first actual demonstration for a specific hockey puck size.

In summmary, as far as we are aware, there has so far been no such demonstration of a DAA system utilizing RID as input for aircraft detection and subsequent collision avoidance functionality. Thus, in addition to our proof-of-concept demonstration of an implementation of RID in a DAA system for drone-to-drone collision avoidance, the newly FAA-mandated RID capabilities of UAS in the United States [2] allow for development of new RID-based DAA systems as well as integration of RID into existing DAA system frameworks and algorithms.

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 detection range (a few hundred meters) and the anticipated collision avoidance perimeter size (few tens of meters). This is not a large margin for collision avoidance. Also, the range is not a part of the ASTM RID specification, so 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 general use case. Drone light shows and drone swarms may be an example where it is appropriate, since relative speed between any two drones is low and 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. And 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 accounted for 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 off-the-shelf components. This system does not rely on any additional hardware other than one ESP32 receive board, which costs around ten U.S. dollars and only weighs 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 would like to 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] "Remote Identification of Drones — Federal Aviation Administration." [Online]. Available: https://www.faa.gov/uas/getting_started/remote_id

[3] "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

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

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

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

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

[8] *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

[9] 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/

[10] 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

[11] 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.

[12] 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.

[13] 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.

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

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] E. Larsson-Kapp, V. Kniivilä, Z. Wang, M. Wzorek, A. Lemetti, and A. Gurtov, "Trust-based collision avoidance for unmanned aircraft systems," in *2024 IEEE International Conference on Aerospace and Signal Processing (INCAS)*. IEEE, 2024, pp. 1–6.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

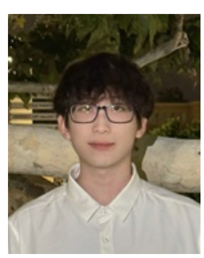
[36] A. D. Evans, M. Egorov, A. Anand, S. E. Campbell, S. Zanlongo, T. Young, and N. Sarfaraz, "Safety assessment of utm strategic deconfliction," in *AIAA Scitech 2023 Forum*, 2023, p. 0965.

[37] 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.

[38] 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.

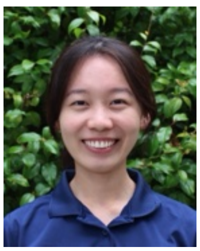
[39] A. Weinert, L. Alvarez, M. Owen, and B. Zintak, "Near midair collision analog for drones based on unmitigated collision risk," *Journal of Air Transportation*, vol. 30, no. 2, pp. 37–48, 2022.

[40] 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.



WENTAO CHEN is currently working toward the B.S. degree in Computer Science and Engineering at the University of California, Irvine, Irvine, CA, USA, and is graduating in 2024. His current research interests include embedded systems, architecture, and security.

AASHAY S. SHAH is currently pursuing the B.S. degree in Computer Science and Engineering at the University of California, Irvine, Irvine, CA, USA, and will be graduating in 2025. His interests include distributed systems, networking, and embedded systems.



MENGJIE XIE is currently working toward the B.S. degree in Computer Science and Engineering at the University of California, Irvine, Irvine, CA, USA, and is graduating in 2024. She will pursue the M.S. degree in Computer Science at Stanford University, Stanford, CA, USA, starting 2024. She is interested in robotics, embedded systems, and machine learning.

PETER J. BURKE (M'02–SM'17-F'20) received the Ph.D. degree in physics from Yale University, New Haven, CT, USA, in 1998. From 1998 to 2001, he was a Sherman Fairchild Postdoctoral Scholar in physics with the California Institute of Technology, Pasadena, CA, USA. Since 2001, he has been a Faculty Member with the Department of Electrical Engineering and Computer Science, University of California at Irvine, Irvine, CA, USA. His current research interests include EECS, BME, chemical and biomolecular engineering, materials science and engineering, and chemical and materials physics.