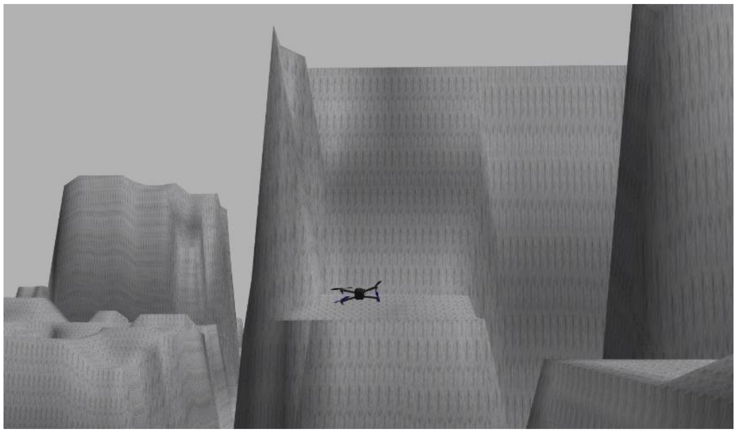
Technologies for Integration of small Unmanned Aerial Systems (s-UAS) in National Airspace System



Prepared by:
Matthew Dechering
Mohammadreza Radmanesh
Manish Kumar

Prepared for:
The Ohio Department of Transportation,
Office of Statewide Planning & Research

State Job Number 135503

July 2018

Final Report

Technical Report Documentation Page

1. Report No. **FHWA/OH-2018-12**	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle **Technologies for Integration of small Unmanned Aerial Systems (s-UAS) in National Airspace System**	5. Report Date **July 2018** 6. Performing Organization Code
7. Author(s) **Matthew Dechering (0000-0002-2634-7913)**	8. Performing Organization Report No.
9. Performing Organization Name and Address **University of Cincinnati** **2600 Clifton Avenue** **Cincinnati, OH 45221**	10. Work Unit No. (TRAIS) 11. Contract or Grant No. **SJN 135503**
12. Sponsoring Agency Name and Address **Ohio Department of Transportation** **1980 West Broad Street** **Columbus, Ohio 43223**	13. Type of Report and Period Covered **Final Report** 14. Sponsoring Agency Code

15. Supplementary Notes

Prepared in cooperation with the Ohio Department of Transportation (ODOT) and the U.S. Department of Transportation, Federal Highway Administration

16. Abstract

Small Unmanned Air Systems (s-UAS) have generated a lot of interest in recent years due to their potential to revolutionize applications in civilian domains. For the widespread use of s-UAS to become a reality for civilian applications, s-UAS must safely and reliably integrated into the National Airspace System (NAS). Because the technology in this field is advancing rapidly, there has been no comprehensive study on the technologies involved. This study seeks to provide a comprehensive survey of the recent technological advances, identify potential issues, and propose solutions. Focus will be on the Sense and Avoid (SAA) and Unmanned Systems Traffic Management (UTM) aspects of s-UAS integration in NAS. This study surveyed available and developing technologies, performed a comparative study of existing solutions proposed by industry, academia, and the government, and formulated a UTM solution for urban package delivery.

17. Keywords **UTM, SAA, NAS, Unmanned Airspace Traffic Management, Sense and Avoid, National Airspace System**	18. Distribution Statement **No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161**

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified	**Unclassified**		

Form DOT F 1700.7 (8-72) **Reproduction of completed pages authorized**

Technologies for Integration of small Unmanned Aerial Systems (s-UAS) in National Airspace System

Prepared by:
Matthew Dechering, University of Cincinnati

July 2018

Prepared in cooperation with the Ohio Department of Transportation
and the U.S. Department of Transportation, Federal Highway Administration

The contents of this report reflect the views of the author(s) who is (are) responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Ohio Department of Transportation or the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

Acknowledgments

Fred Judson

Table of Contents

List of Figures

List of Tables

List of Notations

Acronym	Meaning
ADS-B	Automated Dependent Surveillance-Broadcast
AGL	Above Ground Level
ANSP	Air Navigation Service Provider
ATM	Air Traffic Management
FIMS	Flight Information Management System
GPS	Global Positioning System
HMD	Horizontal Miss Distance
IMU	Inertial Measurement Unit
LIDAR	Light Detection and Ranging
LTE	Long-Term Evolution
NAS	National Airspace System
RADAR	Radio Detection and Ranging
RTK	Real-Time Kinematic
SAA	Sense and Avoid
s-UAS	Small Unmanned Aerial Systems
SWAP	Size, Weight, and Power
TCL	Technical Capability Level
TOF	Fime-Of-Flight
UAM	Urban Air Mobility
UAS	Unmanned Aerial Systems
USS	UAS Service Supplier
UTM	Unmanned Aerial Systems Traffic Management
V2V	Vehicle-to-Vehicle
VMD	Verical Miss Distance
WAVE	Wireless Access in Vehicular Environments

Executive Summary

Small Unmanned Aerial Systems (s-UAS) have generated a lot of interest in recent years due to their potential to revolutionize applications in civilian domains. For the widespread use of s-UAS to become a reality for civilian applications, s-UAS must be safely and reliably integrated into the National Airspace System (NAS). Because the technology in this field is advancing rapidly, there has been no comprehensive study on the technologies involved. This study seeks to provide a comprehensive survey of the recent technological advances, identify potential issues, and propose solutions. Focus will be on the Sense and Avoid (SAA) and Unmanned Aerial Systems Traffic Management (UTM) aspects of s-UAS integration in NAS. The third quarter report covered a survey of existing and emerging technologies in sensing, computing, and communication devices. The fourth quarter report expanded on the previous goal with a survey of literature on existing solutions and a comparative study of existing solutions. This report includes a summary of the quarterly reports and determination of operational requirements for package delivery s-UAS and recommended algorithmic solutions for these operational requirements.

Project Background

While Unmanned Aircraft Systems (UAS) have been used in military applications for many years, UAS, particularly small UAS (s-UAS) have recently generated a lot of interest in civilian domains, due to their potential to revolutionize several applications. Potential areas of application include emergency management, law enforcement, infrastructure inspection, precision agriculture, package delivery, and imaging/surveillance. Additionally, the FAA expects a large amount of growth in number of unmanned flights, between 162% and 432% by 2021 [1]. With the large numbers of drones expected to be in the air (FAA predicts at least 2.75 Million units by 2021, up from 1.10 million units), several challenges are presented in safe operation of UAS in terms of traffic management. Safe separation needs to be maintained between unmanned systems and other aircraft, between unmanned systems and the ground, and between unmanned systems and stationary objects. Safe integration of UAS into the National Airspace (NAS) involves disciplinary areas that include recent technologies (sensing, command, control, and communications) and regulations. The pace at which technology has been emerging and the complexity arising from the close interaction of multiple technological areas present a major challenge in integration of s-UAS in NAS. Because a systematic and thorough study of limitations and capabilities of the emerging technologies in this field is absent, this study aims to study the advances, identify potential issues, and propose new solutions focusing on the "Sense and Avoid" (SAA) and "Unmanned Aircraft Systems Traffic Management" (UTM) aspects of UAS integration in NAS.

To this end, the project includes a survey of existing and emerging technologies to determine immediate and near-term capabilities, a comparative study of existing solutions proposed by industry, academia, and government, development of operational requirements for UTM systems, and recommendations for solutions for these operational requirements. Package delivery applications appear to be a priority because of their prominence as a potential UAS application in the near term. Statewide sense and avoid technologies were also a priority, because of the critical safety role they play. Urban air mobility is the topic that ultimate receives the most attention in this study, because it is a field with much unexplored potential.

Research Context

To achieve the overall objective of this project, we pursued four specific technical goals.

Technical Goal #1: Survey of existing technologies, including sensing, computing, and communication devices.

This goal focuses on an exhaustive market survey on existing state-of-the-art technologies available for achieving SAA and UTM. Limitations and capabilities of each of these technologies have been catalogued in the previous report. Technologies include different onboard sensor modalities, ground-based sensors (radar and cameras), Automated Dependent Surveillance-Broadcast (ADS-B), communication devices and protocols, and computing. Onboard sensors include intruder and obstacle detection (LIDAR, RADAR, vision, thermal vision, Time-Of-Flight (TOF) cameras) and localization and position measurement (GPS, inertial measurement units (IMU)). Communication devices and protocols include cellular networks (LTE), satellite-based communication, and Wireless Access in Vehicular Environments (WAVE). Computing includes the capabilities of onboard computers for data processing and path planning and auto-pilots for lower-level control.

To reach this goal, we performed searches for each technology on the internet and recorded the statistics of each product found. Additionally, a literature survey of advances in each technological area was conducted.

Technical Goal #2: Comparative study of existing solutions proposed by industry, academia, and government.

Several novel architectures for UTM have been proposed that provide details about the command and control concept of operations. These include the Google and Amazon's different Airspace Service Provider concepts [2] [3] [4]. NASA also has many advances and simulations for UTM operations [5]. Table 1 in Appendix A shows NASA's Technical Capability Level Roadmap. At time of writing, TCL 3 is scheduled for the spring of 2018. Rockwell Collins also has made advances with their UAS services. [6]. This specific goal will carry out systematic comparative studies between the solutions provided by industry, government, and academia to evaluate their respective effectiveness and feasibility in terms of technologies available.

To reach this goal, we conducted a literature survey of Technologies listed in NASA's UTM research, as well as major papers published by Google, Amazon, and Rockwell Collins as representatives of industry. Solutions presented by academia had some overlap with papers investigated in the previous goal.

Technical Goal #3: Development of operational requirement for UTM and SAA

There are several potential ODOT applications that require multiple UAS to operate simultaneously. These include, but are not limited to: search and rescue, emergency operations, quick clear, and infrastructure inspection. The objective of this specific goal is coming up with an operational requirement for multi-UAS operation based on one such application. ODOT professionals expressed the most interest in package delivery in urban environments.

To reach this goal, we conversed with our contacts at ODOT to determine the desired area of interest for a specific application. We discussed a metric for measuring a UTM algorithm.

Technical Goal #4: Development of a solution for the UTM and SAA problem with operational requirement identified in Technical Goal #3 and its verification via extensive simulation

The objective of this specific goal is to develop an approach that will allow multiple UAS to operate in a shared airspace in a safe and reliable manner. Realistic ADS-B and GBSAA models developed based on ADS-B sensors available at UC and GBSAA sensors available at the UASTC facility in Springfield, OH. The algorithm development will be achieved via design/allocation of airspace, dynamic geo-fencing, congestion management via path planning and separation management via obstacle avoidance algorithms. This includes developing algorithms aimed at reducing computational effort and increasing the robustness.

To meet this goal, we have developed algorithms based on the previous work done by our own lab and public papers, and we are in the process of testing them.

The FAA defines Sense and Avoid (SAA) as "the capability of a UAS to remain well clear from and avoid collisions with other airborne traffic," in the context of UAS [7]. The UAS ExCom Science and Research Panel (SARP) published a recommendation for s-UAS well clear that involved a 250 ft. Vertical Miss Distance (VMD) and a 2000 ft. Horizontal Miss Distance (HMD). Unlike the recommendations for large UAS well clear, no time-based definitions of well clear were proposed. [8]. Under the FAA's small UAS rule, remote operation is allowable in G-class airspace under 400 ft. AGL, to provide a buffer between s-UAS operations and manned airspace. Following these definitions, traffic in G-class airspace is quite limited. Only two s-UAS can operate within the same vertical column. In uncongested rural areas, these rules would be effective at meeting well-clear requirements while permitting effective use of airspace, but in congested urban environments these could severely limit airspace, especially between buildings taller than 400 ft. that are separated by less than 150 ft. Since such areas would be impassible with current standards, if urban s-UAS become widespread, highly urban areas will likely be limited to highly maneuverable s-UAS. The FAA and several major companies have suggested UTM strategies for various environments. These are all developments towards Urban Air Mobility (UAM), a safe and efficient system for urban UAS services.

Research Approach

In the literature search completed in the third quarter, existing technologies for s-UAS were investigated. Low cost and size, weight, and power (SWAP) ADS-B is currently available, and able to provide coverage in areas problematic to traditional ground-based RADAR [9].ADS-B is vulnerable to electronic attack [10], and the broadcast frequencies will be congested bandwidth under alarge s-UAS traffic load. Because of these limitations, an"ADS-B like" solution is preferred. The standardization and implementation of such a system is a key step for ATM. On-board obstacle sensors come in a variety of shapes and sizes, from technologies that directly measure distance like LIDAR, RADAR, and TOF cameras, to the many types of camera technologies. Examples of both major types of sensors have been shown to be effective in object detection, but one type of sensor has not been standardized. Regardless, sensors should be available that successfully detect and avoid unexpected obstacles. Self-localizing conventional GPS setups exist that can have a horizontal accuracy of 3m and lower. RTK (real-time kinematic) GPS requires a dedicated base station, but can self-localize with an accuracy of up to 1cm. Regardless, s-UAS can be expected to have high positioning accuracy in the near future, especially in rural environments. In urban environments, buildings can function as canyons, denying GPS or inducing accuracy-reducing multipath errors. Therefore, urban s-UAS must have sensors that allow them to navigate in the event of GPS denial. [11] Finally, s-UAS will need to be able to communicate to Airspace Service Provider (ASP) and/or other s-UAS. LTE is a strong technology in urban environments because of its ability to use existing infrastructure. This technology continues to develop, and its exact implementation depends on and influences the decisions with s-UAS communication protocols. Appendix A contains tables comparing various existent technologies examined in this stage.

In the solution study completed in the fourth quarter, a comparison between existing solutions was made. As the airspace becomes more congested, determining permission to access airspace becomes more important. Amazon's best-equipped, best-served model [3] is a straightforward implementation that only allows s-UAS to fly where they can maneuver and perform SAA functions effectively. This will mean that new s-UAS will need to have their capabilities rated, a service which most federal s-UAS test sites already perform. Furthermore, cooperative SAA requires a well-defined communications protocol. Considering the weaknesses of ADS-B, a well-defined ADS-B like system will be one of the next major steps in terms of in terms of inter s-UAS communication. Google's public key infrastructure is one example that would be more secure than the current unencrypted system. [2] With the large amount of data produced by and needed to handle high-volume s-UAS flights, major computational services like Rockwell Collins' WebUAS [6] will play a role in the future of s-UAS management. Airspace monitoring, flight abortion/re-routing, and separation assurance will be key features. Airspace service providers interfacing with the FAA UAS data exchange are the current direction that flight management is headed. They are currently very useful because to the FAA, because it can outsource the automation of flight management to several partners. The challenge will be having all partners acting cooperatively in the same airspace. Not all partners use the same airspace model for obstacles, but they do share the FAA's flight data, which is how traffic is currently managed. Additional information on existing and proposed UTM solutions can be found in Appendix B.

The UTM architecture proposed by NASA is shown in Figure 1. [11] Here, the Air Navigation Service Provider (ANSP) provide traditional air traffic management (ATM) services. UTM is separate from and complementary to UTM; UTM operators and stakeholders conduct and provide support for s-UAS operations independent of the ANSP's scope of influence, but not in isolation of it. Interactions between the two are coordinated by the Flight Information Management System (FIMS), a central cloud-based component that also acts as a broker of information between UTM stakeholders. UAS Service Suppliers (USS) that meet minimum requirements for functionality, quality of service, and reliability. USS then support missions by UAS operators. Connections and communications are internet-based and built on industry standards and protocols.

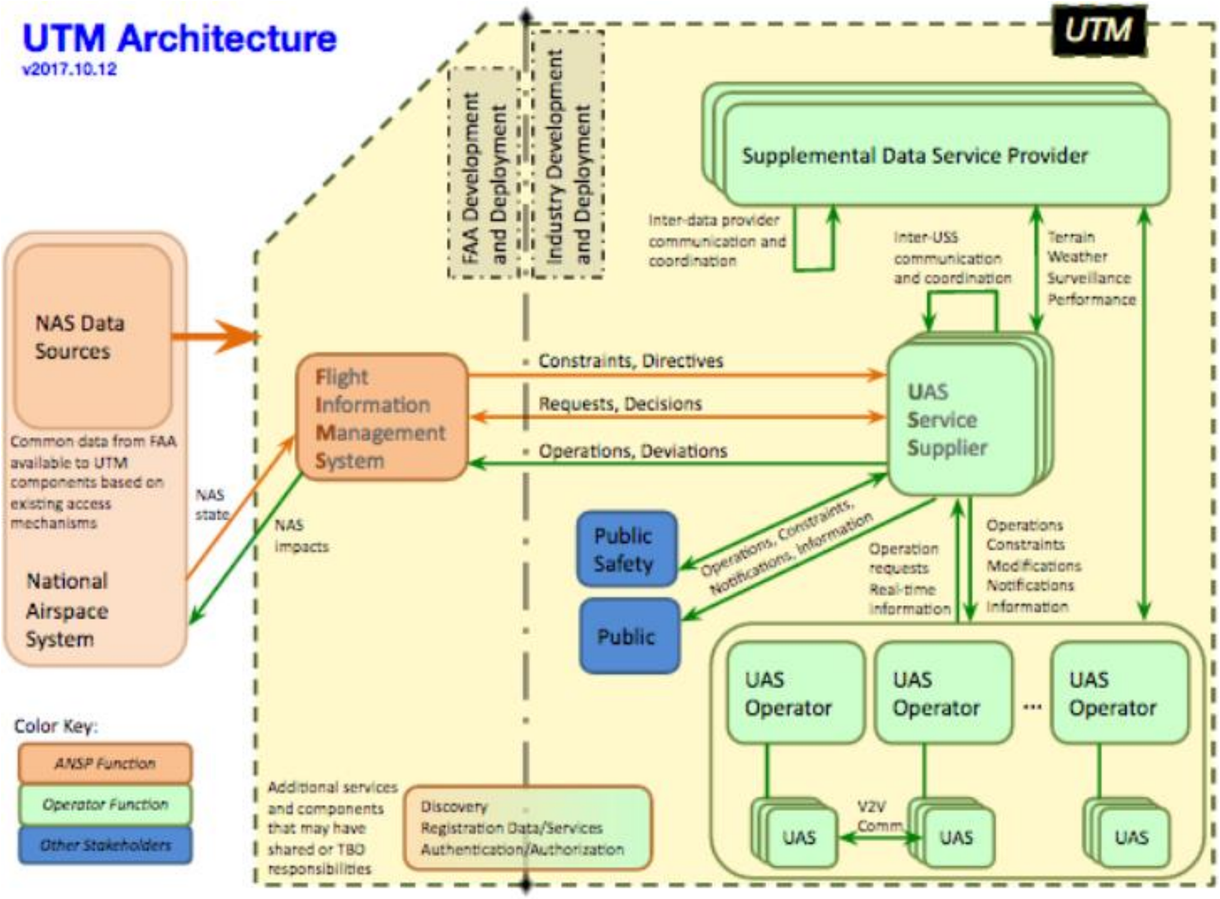

UTM Architecture
v2017.10.12
UTM
FAA Development and Deployment
Industry Development and Deployment
Supplemental Data Service Provider
Inter-data provider communication and coordination
Inter-USS communication and coordination
Terrain
Weather
Surveillance
Performance
NAS Data Sources
Common data from FAA available to UTM components based on existing access mechanisms
NAS state
NAS impacts
National Airspace System
Flight Information Management System
Constraints, Directives
Requests, Decisions
Operations, Deviations
UAS Service Supplier
Public Safety
Public
Operations, Constraints, Notifications, Information
Operation requests
Real-time information
Operations
Constraints
Modifications
Notifications
Information
UAS Operator
UAS Operator
UAS Operator
UAS
UAS
UAS
V2V Comm
Color Key:
ANSP Function
Operator Function
Other Stakeholders
Additional services and components that may have shared or TBD responsibilities
Discovery
Registration Data/Services
Authentication/Authorization

Figure 1: NASA UTM Architecture

With this centralized architecture, Vehicle – to – Vehicle (V2V) communication can play a key role in SAA and conflict resolution. SAA applications often rely on knowledge of an s-UAS's position and velocity and knowledge of intruder position and velocity. [12] More detailed knowledge of intended maneuvers of the intruder are also seen as helpful. [12] ADS-B or an ADS-B like V2V communication system that also provides information on intended maneuvers (i.e. turns) would therefore be the most needed form of V2V communication. In the absence of any V2V communication, SAA must be provided by onboard sensors and/or coordination through possibly multiple USS. For an example of two UAS with separate operators subscribing to separate USS, the coordination would go through communication between the separate USS.

Because of the expedient decrease in the cost of low-SWAP sensors, s-UAS situational awareness is set to grow by leaps and bounds soon. After the initial survey of sensor technologies and UTM solutions, we began investigating algorithms for traffic management. The accuracy of the sensors means completely cooperative vehicles can operate near one another. Intruders still pose a problem, despite some model being equipped with sensors sufficient to anticipate anything that their maneuverability allows them to avoid. Initially, package delivery in an urban environment was considered because large numbers of package delivery s-UAS pose a challenging problem in a sky where traffic between s-UAS is codified to the extent of other vehicles i.e. conventional aircraft or cars. The mission definition was easy to define for package delivery s-UAS. The following algorithms will be effective for package-delivery s-UAS or any s-UAS with a similar mission definition. The mission definition for a package delivery service leaving a depot and travelling to N destinations is: take off from depot, travel to destination 1, perform delivery, travel to the next destination (for n destinations), then return to the depot. Furthermore, the ODOT technical team expressed interest in package delivery as a possible “smart city” or “smart corridor” element.

For the initial algorithm trials, the drone was assumed to be a DJI phantom drone, the obstacle data was taken from a 2007 LIDAR survey of the state of Ohio. Downtown Cincinnati was chosen because it allowed us to prove out avoiding tall buildings relative to ground level while also being able to deal with hills. The initial algorithms were 2D MILP algorithms based on the work of Mohammadreza Radmanesh. [14] [15] An Initial trial showed 10 s-UAS entering an area of airspace at once, which was centered on a tall building. The s-UAS were given priority in first-come, first-served order, and their paths were planned using MILP. Subsequent MILP calls listed the predicted positions of all paths planned using MILP so far. If MILP could not find a solution, the request for airspace was instead delayed until MILP could find a solution. Because MILP suffers performance problems when the temporal path length grows and/or the finite horizon grows, MILP became impractical for planning entire package delivery missions. To solve this, a top-level search algorithm was implemented. RRT* and A* were both considered. The greedy nature of A* worked well, since nodes were placed every 8m, and A* would immediately seek the goal, in contrast to RRT’s rapid expansion in all directions. The pseudo-code for the A* and MILP algorithms at this stage are shown in part 1 of appendix C. The computer used for this simulation was a 64-bit Dell Optiplex 7440_AIO with a dual-core 3.30 GHz Intel i5-6600 CPU and 8 GB Ram. With these algorithms complete, the next step was to move to 3D. MILP becomes more complicated in 3D, with smaller finite horizon and/or max number of time steps possible. The equations for this are shown in part 2 of appendix C. Upon

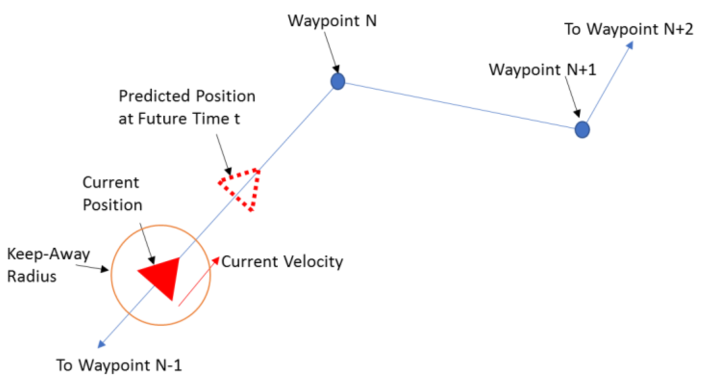


Figure 2: Predicting the Path of the s-UAS

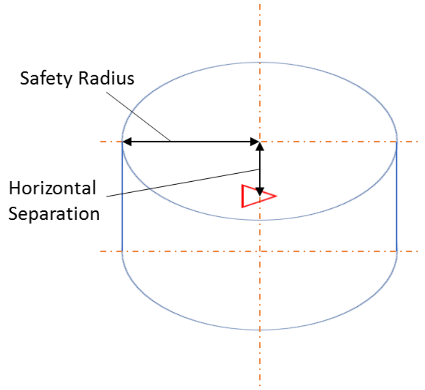


Figure 3: Keep-Away Volume for an s-UAS

seeing the nature the problem could pose at this stage, we made the choice to switch to a higher-level path planner that made resolution waypoints. The MILP algorithm still has its applications: it can be used for a busy intersection such as the 3-D version of [11]. However, the infrastructure is not yet established for this kind of a set-up. The 3D version of A* was used to find initial paths ignorant of traffic, then traffic is managed as in a 3D version of the system presented in [12]. To adapt to higher volumes and/or manned traffic, a market-based solution could be used to further optimize. Additionally, each UAS is prioritized based on remaining expected flight time and battery capacity, which guarantees s-UAS that are trying to leave the airspace have right of way. The pure 3D A* was able to handle 30+ UAS in tests where they moved no more than 8m/s. (the cell size was 8m) The pseudo-code for this algorithm is shown in part 3 of appendix C. The results of this algorithm are summarized in Table 10 in appendix D. The final version of this algorithm is being modified to work with the UC MASTER LAB's GCS software, which can be seen in Figure 6: UC's Flymaster Ground Control Program with auto-generated waypoints.

The sense and avoid algorithm detects conflicts using the current position, current heading and velocity, and planned mission of the s-UAS. For all t in a time window greater than or equal to the stopping time for all vehicles in the airspace, the position is predicted for all s-UAS as in Figure 2. Outside of more complicated commands, a UAS will attempt to reach the next waypoint by following a linear path. A conflict is detected if two or more UASs have their keep-away volumes intersect at a time t. The safety radius and horizontal separation for each s-UAS is determined from the sum of error in position and velocity measurements, the s-UAS's maneuverability, and the expected maximum deviation from the s-UAS's mission trajectory. Each s-UAS has an initial mission entered by a user. Straight lines between waypoints for this

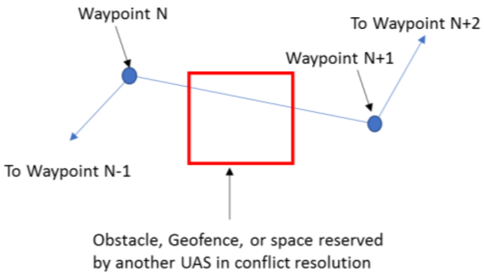

Waypoint N
To Waypoint N+2
Waypoint N+1
To Waypoint N-1
Obstacle, Geofence, or space reserved
by another UAS in conflict resolution

Figure 4: Mission prior to Correction for Obstacles or Conflict Resolution

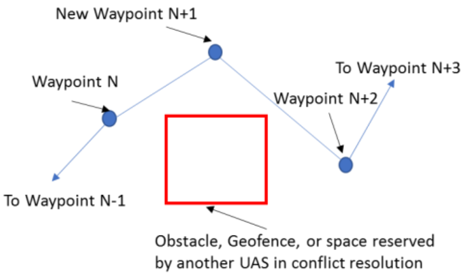
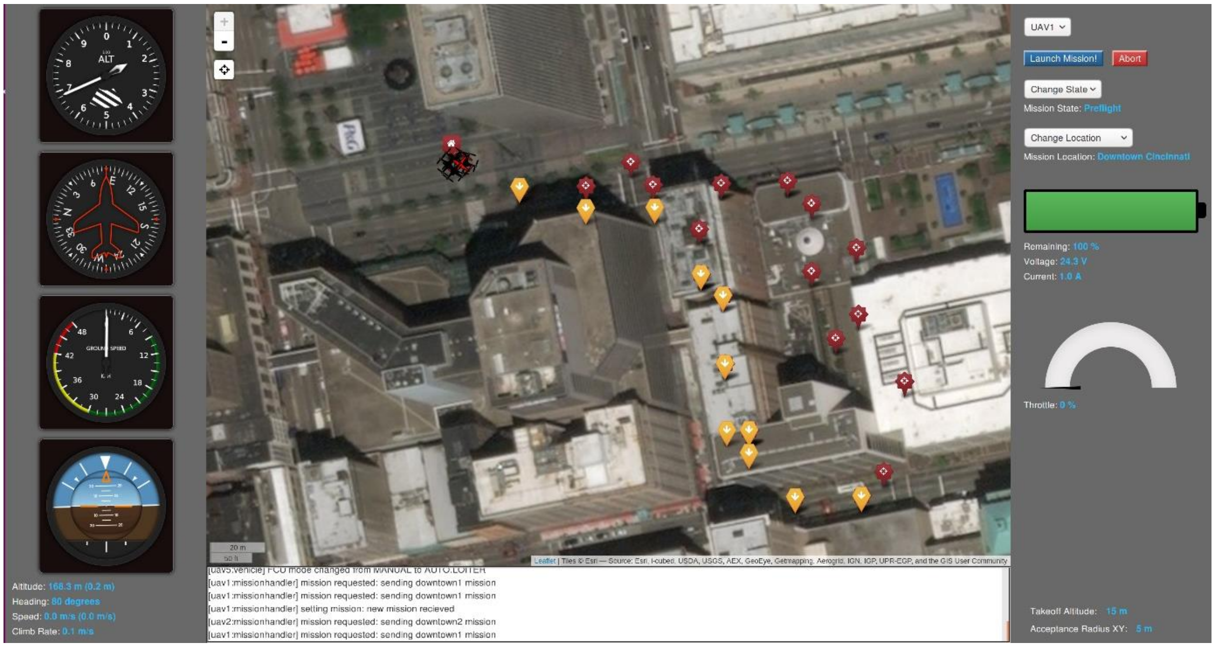

New Waypoint N+1
Waypoint N
To Waypoint N+3
Waypoint N+2
To Waypoint N-1
Obstacle, Geofence, or space reserved
by another UAS in conflict resolution

Figure 5: Goal Path after Initial Routing or Conflict Avoidance

mission may conflict with known obstacles or geofences, as shown in Figure 4. Because of this, for each pair of waypoints the A* conflict avoidance algorithm provides a set of waypoints that form a viable path, as seen in Figure 5. The conflict avoidance algorithm is also called whenever a conflict is detected. Each s-UAS has a priority rating based on its remaining battery life, remaining mission time, and any other factors for mission importance. In a conflict, the lower-priority s-UAS is re-routed to avoid the keep-away volumes of each s-UAS of higher priority.

UC Master LAB's GCS software uses the Pixhawk interface to control multiple UAS meant to bear package/package-like payloads. For the initial tests which will be simulated in Gazebo, the cell size will be 8m and the UAS will move at 1-2m/s. once the initial effectiveness is confirmed these speeds will increase for the Gazebo simulation. Additionally, there will be a live test with 3 UAS controlled from a ground station. The GCS traffic manager will manage the trajectories of the 3 UAS as the maneuver towards goals in such a way as to converge. This test will be run at low speed, so the pilots on-hand can grab control of the s-UAS should anything go wrong. For all tests in gazebo and live, the following conditions will be tested: two converging paths, 3 converging paths, 2 paths that would converge if they did not have a semi-circle bypass. In addition to horizontal set-ups, ascending & descending, ascending & horizontal, and descending & horizontal are also options.


ALT
GROUND SPEED
UAV1
Launch Mission!
Abort
Change State
Change Location
20 m
50 ft
Leaflet | Tiles © Esri — Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community

Mission State: Preflight

Mission Location: Downtown Cincinnati

Remaining: 100 %
Voltage: 24.3 V
Current: 1.0 A

Throttle: 0 %

Altitude: 168.3 m (0.2 m)
Heading: 80 degrees
Speed: 0.0 m/s (0.0 m/s)
Climb Rate: 0.1 m/s

```
[uav5:vehicle] FCU mode changed from MANUAL to AUTO.LOITER
[uav1:missionhandler] mission requested: sending downtown1 mission
[uav1:missionhandler] mission requested: sending downtown1 mission
[uav1:missionhandler] setting mission: new mission recieved
[uav2:missionhandler] mission requested: sending downtown2 mission
[uav1:missionhandler] mission requested: sending downtown1 mission
```

Takeoff Altitude: 15 m
Acceptance Radius XY: 5 m

Figure 6: UC's Flymaster Ground Control Program with auto-generated waypoints

Research Findings and Conclusions

The technology survey largely showed that technology in sensing and computing in s-UAS is expanding rapidly, which is excellent for the fate of situational awareness, but it makes determining proper equipage more difficult simply because of product turnover rate. Most solutions to UTM by NASA have not focused on urban environments as they are beginning TCL 3. However, the FAA has begun to make designations and steps towards automating the access process.

The reconstruction work of Cincinnati resulted in a detailed map, as shown in Figure 7, Figure 8, and Figure 9 in appendix D. MILP is more suited to controlled time intervals and areas, but it plans effective routes when given good data, and can do so for N UAVS in an area. The 2D MILP scenario is shown in Appendix. Each s-UAS takes off from a point, travels to two destinations, then returns to base while avoiding the other vehicles which are trying to do the same thing. Figure 10 and Figure 11 in appendix D show two scenarios where this is the case. The larger 3D A* works effectively for a sparse population of UAVS within an urban volume and works even if there are a large number in close proximity. Figure 12 shows the result of 30 paths, being planned simultaneously. In near gridlock, this form of A* should devolve to a set of right-of-way rules, which would keep traffic flowing.

A* is a tool guaranteed to find the optimal solution, if it exists. If no solution exists (i.e. the goal is in an obstacle) it is good to implement a check, especially if the number of nodes the algorithm could explore is very large. Otherwise A* runs until it exhausts all available nodes. Testing with Gazebo and the PX4 flight stack showed valuable information on relying on GPS for information. In one experiment, an s-UAS lost its GPS fix and wandered from its path for a significant distance before finding it again. Because of this, we modified the prediction element to increase the avoidance radius for the s-UAS in the event of loss of a GPS position fix. There is an opportunity for additional work in urban air mobility in the event of GPS denial. Experiments so far have been limited to avoiding s-UAS of the same package delivery class. Avoiding manned aircraft and fixed wing s-UAS is another opportunity for further research. A* is shown to work quickly enough to be effective in conflict avoidance and re-routing problems and allows for additional optimization algorithms (i.e. a market-based solution) to be built on top for airspaces where a priority system is inefficient.

Recommendations for Implementation of Research Findings

For a ground-based sense and avoid system that provides statewide monitoring of s-UAS traffic, we recommend a network of low-altitude radar sensors to monitor areas where existing radar infrastructure cannot provide coverage. We recommend mandating an ADS-B – like system for BVLOS s-UAS operations. We also recommend each s-UAS carry and obstacle avoidance sensor and the onboard computing capability to avoid terrain, birds, and aircraft not detected by other means. We also recommend each s-UAS carry a communication device through which it interacts with ASP's and/or the ATC as necessary.

For Implementing a statewide s-UAS management system, we suggest determining airspace performance requirements for major urban areas and other areas where s-UAS operations are more challenging or restricted. Performance requirements can include: clearance that s-UAS are expected to maintain with obstacles and each other, maximum size of an s-UAS, maneuverability, and speed limit per segment of airspace within the area.

For traffic management in an airspace that is congested and can be modelled effectively as a set of cells, an A* search algorithm to find an initial path, a method to estimate the path in the immediate future, and application of an intersection resolution algorithm is a good choice. How dedicated airspace management is to an area will naturally be a function of the level of congestion of that area.

On top of the A* search algorithm, an additional conflict avoidance algorithm can be built. There are a wide variety of conflict avoidance solutions, including market-based and bidding-based solutions in addition to the conflict avoidance solutions already developed by NASA and industry partners. Market or bidding-based solutions allow for conflict resolution that does not need to consider the behavior of all vehicles in the airspace, which can be advantageous if information is limited. Additionally, a market-based solution can be implemented to guarantee optimal airspace usage, which is not guaranteed by a priority-based solution.

Bibliography

[1] Federal Aviation Administration, "FAA Aerospace Forecast Fiscal Years 2017-2037," United States Department of Transportation, 2017.

[2] Google Inc., "NASA UTM Documents: Google UAS Airspace System Overview," 2015. [Online]. Available: https://utm.arc.nasa.gov/docs/GoogleUASAirspaceSystemOverview5pager[1].pdf.

[3] Amazon, Inc., "NASA UTM Documents: Determining Safe Access with a Best-Equipped, Best-Served Model for Small Unmanned Aircraft Systems," July 2015. [Online]. Available: https://utm.arc.nasa.gov/docs/Amazon_Determining%20Safe%20Access%20with%20a%20Best-Equipped,%20Best-Served%20Model%20for%20sUAS[2].pdf.

[4] Amazon, Inc., "NASA UTM Documents: Revising the Airspace Model for the Safe Integration of Small Unmanned Aircraft Systems," July 2015. [Online]. Available: https://utm.arc.nasa.gov/docs/Amazon_Revising%20the%20Airspace%20Model%20for%20the%20Safe%20Integration%20of%20sUAS[6].pdf.

[5] J. Rios, D. Mulfinger, J. Homola and P. Venkatesan, "NASA UAS Traffic Management National Campaign," in *Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th*, Sacremento, CA, USA, 2016.

[6] G. Elmasry, D. McClatchy, R. Heinrich and B. Svatek, "Integrating UAS into the Managed Airspace through the Extension of Rockwell Collins' ARINC Cloud Services," in *Integrated Communications, Navigation and Surveillance Conference (ICNS), 2017*, Herndon, VA, USA, 2017.

[7] "Sense and Avoid (SAA) for Unmanned Aircraft Systems (UAS). In Final report of the FAA SAA 2009 sponsored workshop".

[8] A. Weinert, "MIT Air Traffic Control Workshop 2016: Small UAS Well Clear," 8 December 2016. [Online]. Available: https://conferences.ll.mit.edu/atc/sites/default/files/1330-Weinert-2016.12.8%20ATCWorkshop-SUASWellClear.pdf.

[9] Federal Aviation Administration, "Fact Sheet-Automatic Dependent Surveillance-Broadcast (ADS-B)," 13 8 2014. [Online]. Available: https://www.faa.gov/news/fact_sheets/news_story.cfm?newsid=16874. [Accessed 27 10 2017].

[10] D. McCallie, J. Butts and R. Mills, "Security Analysis of the ADS-B implementation in the Next Generation Air Transportation System," *International Journal of Critical Infrastructure Protection*, vol. 4, no. 2, pp. 78-87, 2011.

[11] M. Vasirani and S. Ossowski, "A Market-Inspired Approach for Intersection Management in Urban Road Traffic Networks," *Journal of Artificial Intelligence Research*, vol. 43, pp. 621-659, 2012.

[12] S. Balachandran, C. Munoz and M. Consiglio, "Implicitly Coordinated Detect and Avoid Capability for Safe Autonomous Operation of Small UAS," in *17th AIAA Aviation Technology, Integration, and Operations Conference*, Denver, Colorado, 2017.

[13] Federal Aviation Administration, "UAS data Exchange," Federal Aviation Administration, 7 12 2017. [Online]. Available: https://www.faa.gov/uas/programs_partnerships/uas_data_exchange/. [Accessed 14 12 2017].

[14] AirMap, "airmap.com," 12 December 2017. [Online]. Available: https://www.airmap.com/. [Accessed 12 January 2018].

[15] Project Wing, "Project Wing," 2017. [Online]. Available: https://x.company/projects/wing/. [Accessed 12 January 2018].

[16] Skyward, "skyward.io," 1 January 2018. [Online]. Available: https://skyward.io/. [Accessed 12 January 2018].

Appendix A: Overview of Existing Technologies

Table 1: NASA Technical Capability Levels

Capability 1	Capability 2	Capability 3	Capability 4
• Airspace volume use notification • Over unpopulated land or water • Minimal general aviation traffic in area • Contingencies handles by UAS pilot • Enable agriculture, firefighting, infrastructure monitoring	• Beyond visual line-of-sight • Tracking and low density operations • Sparsely populated areas • Procedures and "rules-of-the-road" • Longer range application	• Beyond visual line-of-sight • Over moderately populated land • Some interaction with manned aircraft • Tracking, vehicle-to-vehicle, internet connected • Public safety, limited package delivery	• Beyond visual line-of-sight • Urban environments, higher density • Autonomous vehicle-to-vehicle, internet connected • Large-scale contingencies mitigation • News gathering, deliveries, personal use

Table 2: ADS-B Products

Product	Refernce	Input Power (W)	Weight (g)	ADS-B in	ADS-B out	Size (mm)	Internal GPS
ping2020	uAvionix	0.5	20	yes	yes	25 x 39 x 12	on ping2020i
ping1090	uAvionix	0.5	20	yes	yes	25 x 39 x 12	on ping1090i
XPS-TR	Sagetech		100	no	yes	89 x 46 x 18	no
XPG-TR	Sagetech		100	no	yes	89 x 46 x 18	yes
MXS	Sagetech	15	150	yes	yes	84 x 64 x 19	available

Table 3: LIDAR

Product	Price	Type	Weight (g)	Range (m)	Power (W)	link
Hokuyo 3D-LIDAR YVT-X002	$4,825	3D Scan	750	50	8.4	[]
Hokuyo UST-10LX	$1,700	Planar Scan	130	30	3.6	[]
Hokuyo UTM-30LX	$4,800	Planar Scan	210	30	8.4	[]
Hokuyo UTM-30LX-EW	$5,290	Planar Scan	210	30	8.4	[]
Hokuyo UTM-30LX-F	$5,000	Planar Scan	210	30	8.4	[]
Hokuyo UXM-30LX-EW	$5,165	Planar Scan	800	30	7.2	[]
Hokuyo UXM-30LXH-EWA	$5,875	Planar Scan	1200	80	7.2	[]
Ibeo LUX	$20,000	Planar Scan	900	200	10	[]
Ibeo LUX 8L	$28,899	Planar Scan	1000	200	10	[]
Ibeo LUX HD	$21,599	Planar Scan	1000	120	10	[]
Ibeo miniLUX	$20,000	Planar Scan	450	40	7	[]
lightware SF40/C	$999	Planar Scan	229	100	4.5	[]
Quanergy M8	$1,000	3D Scan	800	150	15	[]
Quanergy M8-1	$6,100	3D Scan	900	200	18	[]
Quanergy S3	$250	3D Scan	Unknown	150	Unknown	[]
Quanergy S3-Qi	$1,200	3D Scan	Unknown	150	Unknown	[]
RIEGL VQ-480-U	Unknown	Planar Scan	7500	950	55	[]
RIEGL VUX-1UAV	Unknown	Planar Scan	3750	920	60	[]
Scanse Sweep	$349	Planar Scan	120	40	3.25	[]
Spectrolab SpectroScan 3D	Unknown	3D Scan	2018	20	30	[]
Velodyne HDL-32E	$25,000	3D Scan	2000	100	12	[]
Velodyne Puck Hi-res	$8,000	3D Scan	830	100	8	[]
Velodyne Puck LITE	$8,000	3D Scan	590	100	8	[]
Velodyne PUCK VLP-16	$8,000	3D Scan	830	100	8	[]

Table 4: Onboard RADAR

Product	Cost	Range (m)	Accuracy (m)	Weight (g)	Power (W)	Update rate (Hz)	link
Aerotenna µSharp 360°	Unknown	40	0.22	243	2.5	80	[]
Aerotenna µSharp Patch	$500	120	0.22	43	1.25	90	[]
Echodyne MESA SSR	$10,000	750	3.25	1250	45	2	[]
Echodyne MESA-DAA	$10,000	750	3.25	817	35	1	[]
Fortem DAA-R20	Unknown	1500	0.0508	464	60	8	[]
IMST DK-sR-1200e	Unknown	307	0.6	280	4.5	10-200	[]
Integrated Robotics IRIS Sensor	Unknown	66	1.24	360	4.5	3.4	[]

Table 5: Cameras

Name	Supplier	resolution and frame rate	power	SENSOR
FLIR DUO Dual Sensor Thermal Camera	FLIR Systems	1920 X1080	2.2 W 5-26 VDC	Uncooled VOx Microbolometer,
DJI Zenmuse X5S	DJI	20.8 MP		CMOS 4/3"
Raspberry Pi Camera		3280 x 2464	3.7V DC	Sony CCD
Edmund 56-578	edmund	768 x 492	12V DC @ 130 mA	Interlaced CCD
HackHd 1080p	--	4000 x 2250	3.7V DC @ 500 mA	Interlaced CCD
PointGrey BlackFly model BFLY-U3-20S4C-CS	Point Grey	1280 x 1024 @ 60 FPS	5V / 380mA	
PointGrey Flea3 model FL3-U3-13E4C-C	Point Grey	1624 x 1224 @ 15 FPS	5V / 380mA	
e-con Systems' See3CAM_CU130	E-con	4224 x 3156 @ 18 FPS, 1280 x 1080 @ 45 FPS	5V / 380mA	CMOS Image
e-con Systems' See3CAM_CU30	E-con	1920 x 1080 @ 42 FPS	5V / 380mA	CMOS Image
e-con Systems' See3CAMCU50	E-con	1920x1080 @ 30 FPS (uncompressed), 2592x1944 @ 15 FPS (MJPG compressed), 1280x720 @ 60 FPS (MJPG compressed)	5V / 380mA	CMOS Image
e-con Systems' See3CAM_12CUNIR	E-con	1280x720 @ 55 FPS (uncompressed 16-bit grayscale)	5V / 380mA	CMOS Image
e-con Systems' See3CAM_11CUG	E-con	1280x720 @ 30 FPS (uncompressed 16-bit grayscale), 2592x1944 @ 15 FPS (MJPG compressed)	5V / 380mA	CMOS Image
e-con Systems' See3CAM_10CUG_C	E-con	640x480 @ 45 FPS (uncompressed RAW Bayer color), 1280x720 @ 60 FPS (uncompressed grayscale), 1280x960 @ 45 FPS (uncompressed),	5V / 380mA	CMOS Image
e-con Systems' See3CAM_80	E-con	1920x1080 @ 30 FPS (uncompressed)	5V / 380mA	CMOS Image
IDS uEye cameras UI-3241LE	IDS	(1280x1024, 60fps, 8bit mono)	5V / 380mA	CMOS Image
PointGrey Blackfly GigE PoE color camera with CS-mount lens and Global Shutter	Point Grey	(1280x1024, 60fps, 8bit mono)	Power over Ethernet (PoE); or 12 V nominal (5 - 16 V)	CMOS Image
e-con Systems e-CAM130_CUTK1:	E-con	13 MP 4-lane MIPI CSI-2 Camera Module (eg: VGA @ 90FPS or 13MP @ 14FPS)		CMOS Image
e-con Systems e-CAM30_CUTK1:	E-con	3.4MP at 30 fps in uncompressed YUV		CMOS Image
e-con Systems e-CAM40_CUTK1:	E-con	4.0 MP 4-lane MIPI CSI-2 RGB IR Camera Module (eg: VGA @ 330FPS or 4MP @ 40FPS).		CMOS Image
e-con Systems e-CAM80_MI8825_MOD:	E-con	8 MP MIPI CSI-2 sensor (eg: 720p @ 55FPS or 8MP @ 11FPS)		CMOS Image
LI-USB30-IMX185 2.42M USB 3.0 Camera		Resolution: 1952H x 1241V Frame rate: 30fps with 1080P		2.42M pixels CMOS Sensor
FCBEH3300	Sony	1080p/29.97 mode to 720p/59.94, 1,450,000 pixel 20x Zoom HD Color Block Camera, image stabilization		CMOS Image
Color Camera Module FCB-EX1020/EX1020P	Sony		6 to 12 V DC/ 3.0 W	CCD
LI-M034USB3-AF 720p WDR USB 3.0 Camera with 18x Zoom Lens		Aptina MT9M034 1.2M pixels Sensor Active pixel: 1280H x 960V Frame rate: 30fps ,Tamron 18x ZOOM lens	USB 3.0 +5VDC	
Hero4 Session	goPro	1440p30 1080p60 720p100 480p120		
Hero4 Silver	goPro	4k15 2.7k30 1080p60 720p120 480p240		
Hero4 Black	goPro	4k30 2.7k50 1080p80 720p120 480p240		
Boscam TR1 FPV		1440x1080 30FPS		1/3 CMOS 5.0 Mega Pixel

Name	Supplier	resolution and frame rate	power	SENSOR
XAT520	Foxeer	NTSC：656 ×492 /PAL：786 ×576	support 5-24V voltage, suitable for 2S-5S battery	1/2" CMOS
XAT600M	Foxeer	PAL: 976(H) x 494(V);NTSC: 768(H)×494(V)	support 5-22V voltage, suitable for 2S-5S battery	1/3 "Sony Super HAD II CCD + Nextchip 2040 DSP
Night Wolf	Foxeer	768(H)×582(V)	support 5~35V super wide voltage	1/2 Inch” CCD digital image Sensor（Industrial Level） and 10bit high performance image signal processor
ARROW	Foxeer	650TVL(b/w), 600tvl(color) PAL: 976(H) x 494(V); NTSC: 768(H)×494(V)	support 5~35V super wide voltage	1/3" CCD
Monster	Foxeer	1280(H)×960(V)	support 5~35V super wide voltage	1/2.9” CMOS
Logitech Webcam C930e	Logitech	Camera: 3MP Video: 1080P @ 30fps		
Nikon D3330	Nikon	6559 x 3689	Onboard	Nikon DX CMOS
Nikon D5300	Nikon	6559 x 3689	Onboard	Nikon DX CMOS
Nikon Coolpix S7000	Nikon	4608 x 3456	Onboard	Nikon CMOS
e-con Systems : (Tara - USB Stereo Camera)	E-con	WVGA((2*752)x480) at 60fps uncompressed format	5V / 380mA	CMOS Image
Code Laboratories: (DUO MLX)	Code Laboratories	- 56 FPS @ 752x480 - 62 FPS @ 640x480 - 123 FPS @ 640x240 - 240 FPS @ 640x120 - 93 FPS @ 320x480 - 184 FPS @ 320x240 - 360 FPS @ 320x120	~2.5 Watt @ +5V DC	
Stereolabs: (Zed)	StereoLabs	2.2K 15 4416x1242 1080p 30 3840x1080 720p 60 2560x720 WVGA 100 1344x376	5V / 380mA	
IDS: (N10)	IDS	752 x 480 px 1/3" Wide VGA sensor	5V / 380mA	CMOS Image
Leopard Imaging: (LI-USB30)	Leopard	Active pixel: 752H x 480V Frame rate: 30fps	5V / 380mA	Mono CMOS Sensor
Microsoft Kinect Structured-Light 3D Depth camera	Microsoft	30 frames per second (FPS)		
Asus XTion Pro Live RGB-D	ASUS			
Intel® RealSense™ Robotic Development Kit Intel® RealSense™ Developer Kit (SR300)	Intel	720p at 60 frames per second video capture optimized for streaming 1080p at 30 frames per second video capture for video conferencing Approximately 20cm to 150cm (Software optimized in this range)	5V DC-in @ 3A 5.5/2.1mm jack	CSI (4 Mega pixel), Infrared Camera, Infrared Laser Projector, RGB Color Camera, 2nd Gen ASIC
Vue Pro	FLIR	336x256 Resolution with 6.8mm Lens 336x256 Resolution with 9mm Lens 336×256 Resolution with 13mm Lens 640x512 Resolution with 9mm Lens 640x512 Resolution with 13mm Lens 640x512 Resolution with 19mm Lens 30 Hz (NTSC); 25 Hz (PAL)		
FC-Series S	FLIR	D1: 720 × 576 4CIF: 704 × 576 Native: 640 × 512 Q-Native: 320 × 256 CIF: 352 × 288 QCIF: 176 × 144	11-56 VDC	
LI-THERMAL-DEV kit Features	FLIR	320 x 240, 30 or 60fps	12VDC	

Name	Supplier	resolution and frame rate	power	SENSOR
M1 HD EO High Definition zoom FLIR Thermal Imaging Camera for security	SPI infrared	2.3 Megapixel MP HD 1/2” Progressive 60hz real time, no lag sensor, supporting 1080p, 1080i, 720p output, full digital video technology	Wide range of voltage input (10.8-28V)	
M2-D	SPI infrared	1024×768 Auto Imaging Low Light CMOS sensor		
M3-D	SPI infrared			
M4-D	SPI infrared			
M5-D	SPI infrared			
DJI Zenmuse Z30				CMOS, 1/2.8"
DJI X5S	DJI	H.264 C4K: 4096×2160 23.976/24/25/29.97/47.95/50/59.94p @100Mbps 4K: 3840×2160 23.976/24/25/29.97/47.95/50/59.94p @100Mbps 4K: 3840×1572 23.976/24/25/29.97p @100Mbps 2.7K: 2720×1530 23.976/24/25/29.97p @80Mbps 47.95/50/59.94p @100Mbps FHD: 1920×1080 23.976/24/25/29.97p @60Mbps 47.95/50/59.94p @80Mbps 119.88p @100Mbps		CMOS, 4/3” Effective Pixels: 20.8MP
DJI X4S	DJI	H.264 C4K: 4096×2160 23.976/24/25/29.97/47.95/50/59.94p @100Mbps 4K: 3840×2160 23.976/24/25/29.97/47.95/50/59.94p @100Mbps 2.7K: 2720×1530 23.976/24/25/29.97p @80Mbps 47.95/50/59.94p @100Mbps FHD: 1920×1080 23.976/24/25/29.97p @60Mbps 47.95/50/59.94p @80Mbps 119.88p @100Mbps H.264/H.265 100Mbps		CMOS, 1" Effective Pixels: 20 MP

DJI X5R	DJI	4096x2160(23.98p)
		3840x2160(29.97/23.98p)
		2704x1520(30/25P)
		1920x1080(59.94/29.97p)
				CMOS Effective Pixels: 16 MP

DJI X5	DJI	4096x2160(23.98p)
		3840x2160(29.97/23.98p)
		2704x1520(30/25P)
		1920x1080(59.94/29.97p)
				CMOS Effective Pixels: 16 MP

Blackmagic Micro Studio Camera 4K	Blackmagic	Shooting Resolutions 3840 x 2160, 1920 x 1080 Frame Rates HD 1080p23.98, 24, 25, 29.97, 30, 50, 59.97, 60, 1080i50, 59.94, Ultra HD 2160p23.98, 24, 25, 29.97, 30	Battery Type Canon LP-E6 Battery Life Approximately 1 hour 30 minutes Power 12V-20V	
Blackmagic Micro Cinema Camera	Blackmagic	Shooting Resolutions 1920 x 1080 Frame Rates 1080p23.98, 24, 25, 29.97, 30, 50, 59.94, 60	Battery Type Canon LP-E6 Battery Life Approximately 1 hour 30 minutes Power 12V-20V	
Mako	Alliedvision	VGA to 5 Megapixels, Up to 550 fps		CCD and CMOS

Name	Supplier	resolution and frame rate	power	SENSOR
Guppy Pro	Alliedvision	VGA to 5 Megapixels, Up to 123 fps		CCD and CMOS
Manta	Alliedvision	VGA to 9 Megapixels, Up to 125 fps		CCD and CMOS (Sony, CMOSIS)
Stingray	Alliedvision	VGA to 5 Megapixels, Up to 84 fps		Sony CCD
Prosilica GT	Alliedvision	1.3 to 29 Megapixels, Up to 62 fps		CCD and CMOS (Sony, OnSemi, CMOSIS)
Goldeye G and CL	Alliedvision	QVGA and VGA Resolution, Up to 344 fps		InGaAs sensors
B1942	imperx	2 MP HD @54FPS,1940 x 1460		ICX-674
B2020	imperx	4 MP@20FPS,2048 x 2048		KAI-04022
B2021	imperx	4 MP@17FPS,2048 x 2048		KAI-04070
B2041	imperx	4 MP@34FPS,2048 x 2048		KAI-04070
B2740	imperx	6 MP@25.4FPS,2750 x 2200		ICX-694
B3320	imperx	8 MP@10.6FPS,3296 x 2472		KAI-08050
B3340	imperx	8 MP@21FPS,3296 x 2472		KAI-08050
B3440	imperx	9 MP@17.2FPS,3388 x 2712		ICX-814
B4820	imperx	16 MP@4.2FPS,4872 x 3248		KAI-16000
B4821	imperx	16 MP@4.2FPS,4896 x 3264		KAI-16050
B4841	imperx	16 MP@8.8FPS,4896 x 3264		KAI-16050
B4822	imperx	16 MP@4.1FPS,4864 x 3232		KAI-16070
B4842	imperx	16 MP@7.9FPS,4864 x 3232		KAI-16070
B6620	imperx	29 MP@2.4FPS,6576 x 4384		KAI-29050
B6640	imperx	29 MP@4.7FPS,6576 x 4384		KAI-29050
B1310	imperx	1.3 MP,1280 x 960@39FPS		ICX-445
B1410	imperx	1.4 MP,1360 x 1024@30FPS		ICX-285
B1620	imperx	2 MP,1600 x 1200@44FPS		KAI-2020
B1921	imperx	2 MP HD,1920 x 1080@39FPS		KAI-02150
B1922	imperx	2 MP HD,1940 x 1460@25FPS		ICX-674
B2510	imperx	5 MP,2448 x 2050@9.6FPS		ICX-655
B2520	imperx	5 MP,2448 x 2050@16FPS		ICX-625
B2720	imperx	6 MP,2750 x 2200@12.7FPS		ICX-694
B3420	imperx	9 MP,3388 x 2712@8.6FPS		ICX-814

Table 6: Ground Systems

Ground Unit	Company	type	sUAS detection limit	RADAR	ADS-B
Skylight	Gryphon Sensors	non-cooperative detection	10 km	yes	yes
Harrier DSR-200	Detect	non-cooperative detection	4.8 km	yes	yes
SharpEye™ SxV	Kevin Hughes	non-cooperative detection	1 km	yes	
pingStation	uAvionix	ADS-B Station	241 km		yes
FlightHorizon GCS	Vigilant	situational awareness, C2 solution		yes	yes
ODOT ground based detect and avoid	AFRL and ODOT	non-cooperative detection	17nm x 14 nm block of airspace, >500ft. AGL	yes	yes

Table 7: Common Autopilots

Company	AutoPilot	Source Control	Waypoint Navigation	ADS-B Compatibility	Link
3DR	APM 2.8	Open Source	Yes	Yes	[]
	PixHawk Mini	Open Source	Yes	Yes	[]
AeroQuad	AeroQuad v2.2 Kit	Open Source	Optional	No	[]

Company	AutoPilot	Source Control	Waypoint Navigation	ADS-B Compatibility	Link
DJI	A2	Proprietary	Yes	Yes	[]
	Naza-M Lite	Proprietary	Optional	Yes	[]
	Naza-M V2	Proprietary	Yes	Yes	[]
	Wookong	Proprietary	Yes	Yes	[]
Emlid	Navio2	Open Source	Yes	Yes	[]
Erle Robotics	Erle Brain 3	Open Source	Yes	Yes	[]
	Erle-Brain PRO	Open Source	Yes	Yes	[]
Feiyu Tech	FY-40A	Unknown	No	No	[]
	FY-41AP	Unknown	Optional	No	[]
	FY-41AP Lite	Unknown	Unknown	No	[]
	FY-DOS	Unknown	Unknown	No	[]
	Panda2	Unknown	Yes	No	[]
Free Flight	FF Auto Balance Controller	Unknown	No	No	[]
HobbyKing	AfroFlight Naze32 Rev6 Acro	Unknown	Unknown	No	[]
	KK 2.1 HC	Open Source	Unknown	No	[]
Hobbyking / Crius	All In One PRO	Open Source	Optional	No	[]
	MultiWii Lite	Open Source	Unknown	No	[]
	MultiWii SE	Open Source	Optional	No	[]
Holybro	Pixfalcon	Open Source	Yes	Yes	[]
Hoverfly	HoverflyPRO	Unknown	Optional	No	[]
Intel	Intel Aero	Open Source	Yes	Yes	[]
Intrinsyc	Snapdragon Flight Autopilot	Open Source	Yes	Yes	[]
LibrePilot	CopterControl/Atom	Open Source	Unknown	No	[]
	Revolution	Open Source	Unknown	No	[]
MiKroKopter	Flight-Ctrl ME 2.1 Complete	Unknown	Yes	No	[]
mRobotics	Pixracer R14	Open Source	Yes	Yes	[]
MultiWiiCopter	iNav Sirius™ AIR3 F3 SPI	Open Source	Add-on	No	[]
ProfiCNC	Pixhawk 2.1 Cube	Open Source	Yes	Yes	[]
QuadroUFO	UAVX-ARM32 Full Sensors	Open Source	Unknown	No	[]
Range Video	RVOSD 6	Unknown	Yes	No	[]
SmartAP	3.0 Pro	Unknown	Yes	No	[]
	4 Set	Open Source	Yes	No	[]
Viacopter / FlyDuino	AutoQuad v6.6	Open Source	Yes	No	[]

Table 8: WAVE Technologies

Product	Source
WaveCombo	http://www.redpinesignals.com/Modules/Internet_of_Things/WaveCombo_Family/index.php
LocoMate	http://www.aradasystems.com/locomate-mini-obu/
SnapDragon	https://www.qualcomm.com/products/snapdragon/processors/820-automotive

Table 9: LTE Technologies

Product	Source
Snapdragon	https://www.qualcomm.com/products/snapdragon/modems/4g-lte/x16
HUAWEI E3272	http://www.uasvision.com/2017/04/18/controlling-drones-via-lte-network/
AES-ATT-M14A2A-IOT-ADD-G	https://www.avnet.com/shop/us/p/kits-and-tools/development-kits/avnet-engineering-services/aes-att-m14a2a-iot-add-g-3074457345631510175
Quectel Raspberry pi kit	http://sixfab.com/product/3g-4glte-base-shield/

Appendix B: Study of Existing and Proposed Solutions

Amazon's strategy follows the best equipped, best served model. [3]This means a UAV's permission to access an airspace is decided based on how well equipped the UAV is for that airspace. To operate BLOS and/or in the most challenging airspace, a UAV needs five equipage elements: geospatial data for safe separation from known hazards, online flight planning and management, a reliable internet connection, collaborative vehicle-to-vehicle sense and avoid, and non-collaborative sensor-based sense and avoid. To implement this model, Amazon wants to partition the airspace under 500ft. AGL into areas of different required levels of safety. [4]. These partitions are not one-size fits all: a remote area with less air traffic has a lower required level of safety than an urban area. Under 200 feet AGL is the low speed, localized traffic area. It is used for terminal non-transit operations, such as surveying and inspection. Vehicles with low equipage are restricted to areas in this airspace where they meet the required level of safety. Between 200 and 400 feet AGL class G airspace is the High-Speed transit zone. It is meant for transit operations and requires a higher required level of safety. They are allowed access in emergencies only. In addition to an area's required level of safety being dependent on how urban it is, the aviation authorities are also expected to designate predefined low-risk locations, limited areas with a lower required level of safety. Because load on Air Navigation Service Providers (ANSP) is proportional to number of UAVs in the air, amazon wants ANSPs to delegate some of their responsibilities to automation, although they will remain the central offline authority. Amazon envisions several distributed operators controlling fleets that must coordinate with each other via established protocols and services and vehicle-to-vehicle communication.

Google's strategy focusses licensing and information exchange. [2] They discuss operations under 500 feet AGL class G airspace, and hope for future allowances for UAS in lower altitude airport airspace. They expect UAS to give way to manned aircraft via ADS-B, and envision an ADS-B like system for UAS-UAS collision avoidance. They envision Airspace service providers which will license aircraft for operation and provide separation and planning to the UAS via cellular networks. They envision Project Wing having its own service provider, and other operations will have their own service providers. The ASP is the interface between the UAS operators and air traffic control. The ASP maintains a database of Temporary flight restrictions, no-fly zones, weather, obstacles and terrain, traffic, and flight plans. The ASP will also take data from airspace authorities and data sources, including NOAA, FAA, ATCs, and weather data sources. Google emphasizes all pilots, aircraft, and operators having some form of traceable ID (pilot license, aircraft registration, or operator registration). A UAS ID system would be scalable, allows for authentication, and gives traceability. It wants to do this using the proven public key infrastructure (PKI). The PKI is a security process where a participant creates a public/private key pair and shares one with the registration authority. The registration authority verifies the ID of the participant and provides this verification to the certificate authority. The certificate authority uses this information to provide the participant with a signed certificate, which is a secure encapsulation of the participant's identification data. This certificate will then be used whenever a participant submits a flight plan request to the ASP, who verifies the certificate with a verification authority. The ASP signs off on the flight plan (assuming no conflicts exist). The Airspace participant can now use this signed plan for operations. For the future, google wants ASPs to be allowed to operate as manned aviation does today in uncontrolled G-Class airspace. ASPs need to be open and collaborative with each other and air traffic control authorities. Google also wants the ADS-B ruling for 2020 amended to also apply to helicopter flying below 500 feet AGL over populated areas, as these are the manned aircraft s-

UAS will encounter most often. The public key infrastructure is one step that is more secure than the current unsecured ADS-B communications.

Considering the expansion of s-UAS in the near future, Rockwell Collins speaks about its WebUAS™ services, which are a set of secure cloud-based services for UAS operators to interface together and with air traffic control. [6] It emphasizes computational engines which are separate from the services infrastructure, allowing an operator to select the engine elements – computational engines and services – they need to operate well when setting up a server. It supports FAA System Wide Information Management, and is flexible to further standardization. WebUAS supports third-party engines because its computational engines are separate from its infrastructure, and dedicates servers to fulfill specific needs. WebUAS uses AviNet (a secure global network for airlines and airports) to interface to FAA systems, and uses the same FAA approved firewalling it uses with all aviation customers. WebUAS iis based on national centralized servers that are peers to and under the same constraints as Aircraft Situation Display servers. It will support future expansion by adding regional servers. WebUAS has dedicated servers for ATC tower interfaces, other industry partners, and large operators of UTM. WebUAS supports a number of services for BLOS operations. WebUAS is able to generate many service packages by combining a wide selection of specialized servers and computational engines. It includes a flight plan authorization engine that is able to make recommendations instead of simply accepting or rejecting the plan. A list of engine elements available for service customization include: a measure of how well airspace is monitored, the ability to abort or reroute flights in emergencies, real-time separation assurance, specific industry requirements, collaborative decision making for all types of service providers, graphic depiction of recommended routes, and a measure of the proactive prediction and warning capabilities of the engine.

NASA, Google, and Amazon all refer to an Airspace Service Provider (ASP) in their models. An ASP is a company that provides separation and planning services and interface with FAA data. The beginnings of ASP's are seen in the FAA UAS data exchange, an umbrella agency for partnerships between the FAA and industry to facilitate the sharing of airspace data. [13] Its first partnership is the Low Altitude Authorization and Notification Capability (LAANC), an application through which the FAA may authorize operations under the small UAS rule. It allows operators to interact with maps and provide automatic notification and requests to the FAA. Users can apply through a part 107 process, or through an approved UAS service provider. Approved UAS service providers include AirMap, Project Wing, and Skyward.

AirMap allows users to plan flights with a phone or web app. [14] Paths may be constructed from points, lines, and areas. The app also provides a list of keep-out zones to make low altitude flight planning easy. AirMap does not include topographic information or building information, but does allow users to set the height of their flights. Project Wing claims to safely manage complex flight paths across multiple drones, and was tested with drone delivery to residential yards. The Project Wing UTM platform ensures a route that is clear from buildings, terrain, obstacles and participating aircraft. [15] Skyward has an interactive airspace map that allows users to view flight restrictions and mark hazards. [16] They also have a map of elevations and obstacles driven by LATAS flight planning software. For all of these UAS service providers, communication will be highly important as the skies become more congested. For now, communication is done through the LAANC posting flight notifications.

Appendix C: Algorithm Summaries

Part 1: 2D MILP pseudo code

Notations:
Obstacles are represented by $8m \times 8n$ meter rectangles, where m and n are nonzero integers. Each obstacle has minimum and maximum x and y coordinates. Each vehicle has a minimum and maximum speed, a mass, and a maximum acceleration. Each vehicle submits a flight plan at a specified time with a start point, end-point, and K destinations, where K is a nonnegative integer. There are two major parts to the algorithm: an A* portion that conducts macro path-planning considering only obstacles and a MILP portion that finds an optimal path considering the waypoints from A*, time, vehicle dynamics, obstacle positions, and the other vehicles in the airspace.

PseudoCode for the overarching algorithm is as follows:

```
For each flight plan (from earliest takeoff to latest takeoff) {
    For each pair of waypoints in the flight plan {
        Path = Astar(pair of waypoints) }
    Construct the waypoints in path into segments of short enough length for the s-UAS to traverse in the time given for a MILP call.
    Trajectory = empty set of x, y, t points
    For each segment {
        Segment_Trajectory = MILP( waypoints in segment, trajectories of intruders at current time)
        Trajectory.append(Segment_Trajectory)
    }
    Add the new trajectory to the list of trajectories
}
```

PseudoCode for the A* portion of the algorithm follows a typical A* approach.

Nodes are snapped to a grid with 8m spacing and given added to the closed set if they are too close to an obstacle. This grid is kept consistent between calls so there is no need to recalculate whether a node is valid each time. Furthermore, the grid can be precomputed given a visibility criteria to determine which nodes are valid neighbors. A node is not a valid neighbor to a given node if they are farther apart than the visibility criteria or if a line between them does not pass through an obstacle. The initial and goal positions are snapped to the closest points in the grid that are not in an obstacle.

Pseudocode:

```
Initialize the Start_Node with starting coordinates, no parent, and g and f scores of 0
Initialize all other Nodes with no parent and g and f scores of infinity
Open_set = new priorityQueue
Closed_set = new set
Open_set.add(Start_Node)
```

```
Closed_set.add(nodes from grid that are not valid)
While not_empty(Open_set) {
    Best_node = Open_set.pop()
    Closed_set.add(Best_node)
    If Best_node.coordinates == goal coordinates {
        Path = new set
        Current_node = Best_node
        While Current_node has a parent {
            Path.prepend(Current_node)
            Current_node = parent(Current_node)
        }
        Return Path
    }
    For each neighbor of Best_node {
        If neighbor not in Closed_set {
            tentativeGScore = Best_node.gScore + distance(Best_node, neighbor)
            if tentativeGScore < neighbor.gScore {
                neighbor.gScore = tentativeGScore
                neighbor.fScore = neighbor.gScore + distance(neighbor, goal coordinates)
                parent(neighbor) = Best_node
                Open_set.update(neighbor)
            }
        }
    }
}
Return path unavailable message
```

MILP functions by solving a set of mixed-integer linear programming equations with the Gurobi package. When called, it is given a time during which the s-UAS is expected to traverse a waypoint or set of waypoints from its initial state and a finite horizon it is expected to stay within during this timeframe, in addition to vehicle, intruder trajectory, and obstacle information.

The first constrain is the initial constraint:

$$x_1 = x_{initial}$$
$$y_1 = y_{initial}$$
$$v_{x1} = v_{x,initial}$$
$$v_{y1} = v_{y,initial}$$

Where $x_{initial}, y_{initial}, v_{x,initial}, v_{y,initial}$ are all provided by the flight plan or a previous iteration of MILP. x_1, y_1, v_{x1}, v_{y1} are all of the form x_n, y_n, v_{xn}, v_{yn}, where n is the nth timestep, here equal to 1. The finite horizon constraint is given by:

$x_{initial} - L_{horizon} \leq x_n \leq x_{initial} + L_{horizon} \; \forall \, n \in N_{steps}$

$y_{initial} - L_{horizon} \leq y_n \leq y_{initial} + L_{horizon} \, \forall \, n \in N_{steps}$
$N_{steps} = \{n \in Z^+ | n \leq ceil(\frac{T}{T_d})\}$
Where $L_{horizon}$ is the length of the finite horizon and T is the length of time the segment is to be completed in.

The constraints for the dynamic model are given by:

$$S_{n+1} = A_d S_n + B_d F_n \, \forall \, \{n \in N | n < ceil\left(\frac{T}{T_d}\right)\}$$

$$S_n = \begin{bmatrix} x_n \\ y_n \\ v_{xn} \\ v_{yn} \end{bmatrix}$$

$$A_d = \begin{bmatrix} 1 & 0 & T_d & 0 \\ 0 & 1 & 0 & T_d \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_d = \begin{bmatrix} \frac{T_d^2}{m} & 0 \\ 0 & \frac{T_d^2}{m} \\ \frac{T_d}{m} & 0 \\ 0 & \frac{T_d}{m} \end{bmatrix}$$

$$F_n = \begin{bmatrix} f_{xn} \\ f_{yn} \end{bmatrix}$$

Where m is the mass of the vehicle, f_{xn} and f_{yn} are the force of the vehicle in the x and y directions, respectively.
Acceleration and velocity constraints are given by:

$$f_{xn} \sin\left(\frac{2\pi h}{H}\right) + f_{yn} \cos\left(\frac{2\pi h}{H}\right) \leq f_{\max} \, \forall n \in N_{steps}, h \in Z^+, h \leq H$$

$$v_{xn} \sin\left(\frac{2\pi h}{H}\right) + v_{yn} \cos\left(\frac{2\pi h}{H}\right) - v_{\min} \leq (v_{\max} - v_{min})\left(1 - b_{speed,h,n}\right) \forall n \in N_{steps}, h \in Z^+, h \leq H$$

$$v_{xn} \sin\left(\frac{2\pi h}{H}\right) + v_{yn} \cos\left(\frac{2\pi h}{H}\right) - v_{min} \geq \eta + (-v_{min} - v_{max} - \eta) b_{speed,h,n} \forall \, n \in N_{steps}, h \in Z^+, h \leq H$$

$$\sum_{h=1}^{H} b_{speed,h,n} \leq H - 1 \, \forall n \in N_{steps}$$

Where f_{max} is the maximum force the vehicle can generate, H is an integer chosen to discretize the circle, v_{min} and v_{max} are the minimum and maximum velocity of the vehicle, respectively, $b_{speed,h,n}$ is the binary variable for velocity at time step n in direction h, and η is a very small number. If the minimum velocity is zero, as in multicopter vehicles, the binary variables $b_{speed,h,n}$ can be excluded, and the constraint on velocity becomes:

$$v_{xn}\sin\left(\frac{2\pi h}{H}\right)+v_{yn}\cos\left(\frac{2\pi h}{H}\right)\leq v_{max}\ \forall n\in N_{steps},h\in Z^{+},h\leq H$$

The constraints for arriving at the waypoints and dwelling for a specified amount of time are given by:

$$x_n-x_w\leq r_{min}+M_{big}\left(1-b_{w,n}\right)\forall n\in N_{steps}\ ,w\in Z^{+},w\leq W$$

$$y_n-y_w\leq r_{min}+M_{big}\left(1-b_{w,n}\right)\forall n\in N_{steps},w\in Z^{+},w\leq W$$

$$x_n-x_w\leq -r_{min}-M_{big}\left(1-b_{w,n}\right)\forall n\in N_{steps},w\in Z^{+},w\leq W$$

$$y_n-y_w\leq -r_{min}-M_{big}\left(1-b_{w,n}\right)\forall n\in N_{steps},w\in Z^{+},w\leq W$$

$$b_{w,1}=b_{TOA,w,1}\ \forall w\in Z^{+},w\leq W$$

$$b_{w,n}=\left[\sum_{t=1}^{n}b_{TOA,w,t}\right]-\sum_{t=1}^{n-1}b_{TOD,w,t}\ \forall n\in N_{steps},w\in Z^{+},w\leq W$$

$$\left[\sum_{t=1}^{\mathrm{N_{MAX}}}n(b_{TOD,w,t})\right]-\left[\sum_{t=1}^{N_{MAX}}n\left(b_{TOA,w,t}\right)\right]\geq W_{dwell,w}\ \forall w\in Z^{+},w\leq W$$

$$\sum_{t=1}^{N_{MAX}}b_{TOA,w,t}=1\forall w\in Z^{+},w\leq W$$

$$\sum_{t=1}^{N_{MAX}}b_{TOD,w,t}=1\ \forall w\in Z^{+},w\leq W$$

$$N_{MAX}=ceil\left(\frac{T}{T_d}\right)$$

$$r_{min}=\frac{v_{min}^{2}}{\frac{f_{max}}{m}}$$

$$\sum_{t=1}^{n}b_{TOA,w+1,t}\leq\sum_{t=1}^{n}b_{TOD,w,t}\ \ \forall n\in N_{steps},w\in Z^{+},w\geq W$$

x_w and y_w are the coordinates of waypoint w. M_{big} is a number that is large with respect to the finite horizon. $b_{w,n}$ is the binary variable for being at waypoint w at time step n. W is the number of waypoints. $b_{TOA,w,n}$ and $b_{TOD,w,n}$ are binary variables for arriving at and departing from waypoint w at time step n, respectively. $W_{dwell,w}$ is the amount of time the vehicle is instructed to dwell at waypoint w.

To prevent the vehicle's path from intersecting with obstacles or intruders between time steps, a safety factor is used. The Constraints on this safety factor are:

$$-S_{f,x,n}\leq v_{xn}T_d+\frac{f_{xn}T_d^2}{2m}+\sigma\leq S_{f,x,n}\ \forall\ n\in N_{steps}$$

$$-S_{f,y,n}\leq v_{yn}T_d+\frac{f_{yn}T_d^2}{2m}+\sigma\leq S_{f,y,n}\ \forall n\in N_{steps}$$

Where σ is the minimum safe distance allowed between the vehicle and an obstacle or intruder. $S_{f,x,n}$ and $S_{f,y,n}$ are the safety factors in the x direction and y direction, respectively. The equations to prevent obstacle collisions are:

$$x_n \leq x_{min,o} - S_{f,x,n} + M_{big} b_{obs,o,n,1} \ \forall n \in N_{steps}, o \in Z^+, o \leq O$$
$$-x_n \leq -x_{max,o} - S_{f,x,n} + M_{big} b_{obs,o,n,2} \ \forall n \in N_{steps}, o \in Z^+, o \leq O$$
$$y_n \leq y_{min,o} - S_{f,y,n} + M_{big} b_{obs,o,n,3} \ \forall n \in N_{steps}, o \in Z^+, o \leq O$$
$$-y_n \leq -y_{min,o} - S_{f,y,n} + M_{big} b_{obs,o,n,4} \ \forall n \in N_{steps}, o \in Z^+, o \leq O$$
$$\sum_{k=1}^{4} b_{obs,o,n,k} \leq 3 \ \forall n \in N_{steps}, o \in Z^+, o \leq O$$

Where $x_{min,o}$, $x_{max,o}$, $y_{min,o}$, and $y_{max,o}$ are the minimum x, maximum x, minimum y, and maximum y of obstacle o, respectively. $b_{obs,o,n,k}$ with k = 1, 2, 3, and 4 are the binary variables for being outside of the minimum x, maximum x, minimum y, and maximum y, respectively, of obstacle o at time step n. O is the total number of obstacles in the finite horizon. A similar set of constraints is used to construct the intruder avoidance.

$$x_n \leq x_{min,i,n} - 2S_{f,x,n} + M_{big} b_{i,1,n} \ \forall n \in N_{steps}, i \in Z^+, i \leq I$$
$$-x_n \leq -x_{max,i,n} - 2S_{f,x,n} + M_{big} b_{i,2,n} \ \forall n \in N_{steps}, i \in Z^+, i \leq I$$
$$y_n \leq y_{min,i,n} - 2S_{f,y,n} + M_{big} b_{i,3,n} \ \forall n \in N_{steps},, i \in Z^+, i \leq I$$
$$-y_n \leq -y_{max,i,n} - 2S_{f,y,n} + M_{big} b_{i,4,n} \ \forall n \in N_{steps},, i \in Z^+, i \leq I$$
$$\sum_{k=1}^{4} b_{i,k,n} \leq 3 \ \forall n \in N_{steps}, i \in Z^+, i \leq I$$

Here $x_{min,i,n}, x_{max,i,n}, y_{min,i,n},$ and $y_{max,i,n}$ are the minimum x, maximum x, minimum y, and maximum y, respectively, of intruder i at time step n. $b_{i,k,n}$, with k= 1,2,3, and 4, are binary variables for excluding the vehicle from the area covered by intruder i at time step n. I is the maximum number of intruders in the timeframe and finite horizon. The cost function for this 2D maneuver has acceleration and time components. The time component is given by:

$$J_{time} = \sum_{w=1}^{W} \sum_{n=1}^{N_{MAX}} (n-1) b_{TOD,w,n}$$

The acceleration component is given by:

$$J_{force} = \sum_{k=1}^{2} \sum_{n=1}^{N_{MAX}} W_{slack,k,n}$$

Where $W_{slack,k,n}$ is a slack variable constrained by:

$$f_{xn} \leq W_{slack,1,n} \ \forall n \in N_{steps}$$
$$-f_{xn} \leq W_{slack,1,n} \forall n \in N_{steps}$$
$$f_{yn} \leq W_{slack,2,n} \forall n \in N_{steps}$$
$$-f_{yn} \leq W_{slack,2,n} \ \forall n \in N_{steps}$$

Part 2: 3D MILP pseudo-code

The 3D MILP equations were, in general, similar to their 2D counterparts. The dynamic equations were given by:

$$S_{n+1} = A_d S_n + B_d F_n \; \forall n \in N_{steps}, n \leq N_{MAX}$$

where

$$A_d = \begin{bmatrix} 1 & 0 & 0 & T_d & 0 & 0 \\ 0 & 1 & 0 & 0 & T_d & 0 \\ 0 & 0 & 1 & 0 & 0 & T_d \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B_d = \begin{bmatrix} \frac{T_d^2}{2m} & 0 & 0 \\ 0 & \frac{T_d^2}{2m} & 0 \\ 0 & 0 & \frac{T_d^2}{2m} \\ \frac{T_d}{m} & 0 & 0 \\ 0 & \frac{T_d}{m} & 0 \\ 0 & 0 & \frac{T_d}{m} \end{bmatrix}$$

$$S_n = \begin{bmatrix} x_n \\ y_n \\ z_n \\ v_{xn} \\ v_{yn} \\ v_{zn} \end{bmatrix}$$

The acceleration and velocity constraints are:

$$V_n^T \zeta_{h,j} \leq v_{max} \; \forall n \in N_{steps}, \{ h \in Z | 0 \leq h < H\}, \{j \in Z | 0 \leq j < \frac{H}{2}\}$$

$$F_n^T \zeta_{h,j} = f_{max} \; \forall n \in N_{steps}, \{ h \in Z | 0 \leq h < H\}, \{j \in Z | 0 \leq j < \frac{H}{2}\}$$

Where

$$V_n = \begin{bmatrix} v_{xn} \\ v_{yn} \\ v_{zn} \end{bmatrix}$$

$$\zeta_{h,j} = \begin{bmatrix} \cos(\theta_h)\cos(\phi_j) \\ \sin(\theta_\mathrm{h})\cos(\phi_j) \\ \sin(\phi_j) \end{bmatrix}$$

$$\theta_h = \frac{2\pi h}{H}$$

$$\phi_j = \frac{2\pi j}{(\frac{H}{2} - 1)} - \frac{\pi}{2}$$

Unlike the obstacles in the 2D version, the 3D version uses a grid of terrain heights.
The first 3 constraints determine whether the vehicle is over a grid square:

$$-\frac{L_{grid}}{2} b_{grid,p,n} - M_{big,x}\left(1 - b_{grid,p,n}\right) \leq x_n - x_{grid,p}$$
$$\leq \frac{L_{grid}}{2} b_{grid,p,n} + M_{big,x}\left(1 - b_{grid,p,n}\right)$$
$$\forall n \in N_{xtep}, p \in [1, P]$$
$$-\frac{L_{grid}}{2} b_{grid,p,n} - M_{big,y}\left(1 - b_{grid,p,n}\right) \leq y_n - y_{grid,p}$$
$$\leq \frac{L_{grid}}{2} b_{grid,p,n} + M_{big,y}\left(1 - b_{grid,p,n}\right)$$
$$\forall n \in N_{xtep}, p \in [1, P]$$
$$1 \leq \sum_{p=1}^{P} b_{grid,p,n} \leq 4 \; \forall n \in N_{step}$$

Where L_{grid} is the length of a grid. $M_{big,x}$ and $M_{big,y}$ are just larger than the x and y horizons, respectively. $b_{grid,p,n}$ is whether the vehicle is over grid cell p at time step n. $x_{grid,p}$ and $y_{grid,p}$ are the x and y coordinates of the center of grid cell p. P is the total number of grid cells.
The vehicle is kept above the terrain by a safety factor

$$z_n - z_{grid,low,p} \geq b_{grid,p,n} S_{f,z} - M_{big,z}\left(1 - b_{grid,p,n}\right) \forall n \in N_{step}, p \in [1, P]$$
$$-z_n + z_{grid,high,p} \geq \; b_{grid,p,n} S_{f,z} - M_{big,z}\left(1 - b_{grid,p,n}\right) \forall n \in N_{step}, p \in [1, P]$$

Where $z_{grid,low,p}$ and $z_{grid,high,p}$ are the lower and upper limits of grid cell p, respectively. $S_{f,z}$ is the safety factor in the z direction.
To find the cost, the slack variables for force were calculated similarly to the 2D version:

$$J_f = \sum_{n=1}^{N_{MAX}} \sum_{k=1}^{3} W_{slack,n,k}$$

Where $W_{slack,n,k}$ is the slack variable constrained by:

$$f_{xn} \leq W_{slack,n,1} \; \forall n \in N_{steps}$$
$$-f_{xn} \leq W_{slack,n,1} \forall n \in N_{steps}$$
$$f_{yn} \leq W_{slack,n,2} \; \forall n \in N_{steps}$$
$$-f_{yn} \leq W_{slack,n,2} \forall n \in N_{steps}$$
$$f_{zn} \leq W_{slack,n,3} \; \forall n \in N_{steps}$$
$$-f_{zn} \leq W_{slack,n,3} \forall n \in N_{steps}$$

Initial and final conditions were specified in a similar manner to the 2D version, but as a trial run the start and endpoint were simply fixed.

$$S_1 = \begin{bmatrix} x_{initial} \\ y_{initial} \\ z_{initial} \\ v_{x,initial} \\ v_{y,initial} \\ v_{z,initial} \end{bmatrix} \qquad S_{N_{MAX}} = \begin{bmatrix} x_{final} \\ y_{final} \\ z_{final} \\ v_{x,final} \\ v_{y,final} \\ v_{z,final} \end{bmatrix}$$

At this stage, the 3D MILP algorithm had significant slowdown at the 30 timestep mark used by the 2D algorithm. The 3D algorithm seemed to run in similar time when given only 10 timesteps. Because this level even struggled to trajectory plan in the absence of intruders with the same number of time steps as the 2D algorithm with intruders, the 3D MILP model was abandoned for large scale trajectory planning as being far more intricate than was necessary to accommodate large scale trajectory planning. The 3D MILP algorithm has much more potential to shine in an intersection scenario, where obstacles are nearly constant and the motion of s-UAS through space is all that needs to be managed.

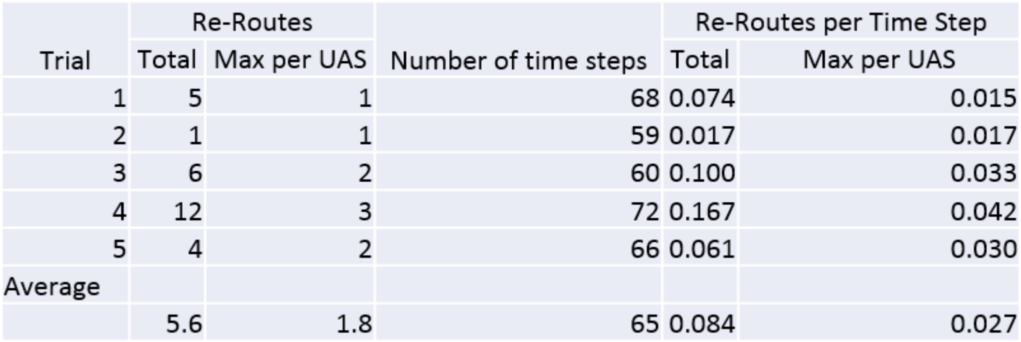
Part 3: Pseudocode for 3D A* with re-routing

A 3-dimensional array of cells is set up prior to A* being run. Each cell is labelled accessible if empty and inaccessible if not.

```
emptyTerrain = setupTerrainModel()
For each s-UAS {
    path(s-UAS) = A*(start(s-UAS),end(s-UAS),emptyTerrain)
    current_Position(s-UAS) = start(s-UAS)
    Current_Goal(s-UAS) = path(s-UAS,goal =2)
}
Time_step = 0
While all paths are not complete {
    reservedTerrain = emptyTerrain
    For each s-UAS {
        If path(s-UAS) not complete {
            Update the current_position(s-UAS) according to the prevous velocity
            If current_position(s-UAS)==end(s-UAS) {
                Path(s-UAS) is complete
            }
            elseIf current_position(s-UAS) == current_goal(s-UAS) {
                Update current_goal(s-UAS) to the next goal in path(s-UAS)
            }
            Update the s-UAS's remaining flight time
            Determine the s-UAS's priority based on its nearness to its final goal and its remaining flight time
            Update the position in reservedTerrain occupied by the s-UAS to inaccessible
        }
        Else {
            Remove the s-UAS
        }
    }
    For each s-UAS, in order of priority {
        Calculate desired velocity based on current_position(s-UAS) and goal(s-UAS)
        temporaryTerrain = emptyTerrain for all spaces not neighbors of current_position(s-UAS)
        temporaryTerrain = reservedTerrain for all spaces that are neighbors of current_position(s-UAS)
        Predict future position if desired velocity is maintained
        If future position is inaccessible in the temporaryTerrain{
            ReRoute = A*(current_position(s-UAS), end(s-UAS), temporaryTerrain)
            Modify path(s-UAS) to indicate the re-Route
            Update current_goal(s-UAS)
            Update desired velocity of s-UAS
            Predict future position of s-UAS
        }
        Update the future position of s-UAS in reservedTerrain to inaccessible
    }
}
```

Appendix D: Algorithm Results

Table 10: A With Re-Routing Results for 30 s-UAS in 160m X 160m Areas of Downtown Cincinnati*



	Re-Routes			Re-Routes per Time Step	
Trial	Total	Max per UAS	Number of time steps	Total	Max per UAS
1	5	1	68	0.074	0.015
2	1	1	59	0.017	0.017
3	6	2	60	0.100	0.033
4	12	3	72	0.167	0.042
5	4	2	66	0.061	0.030
Average					
	5.6	1.8	65	0.084	0.027

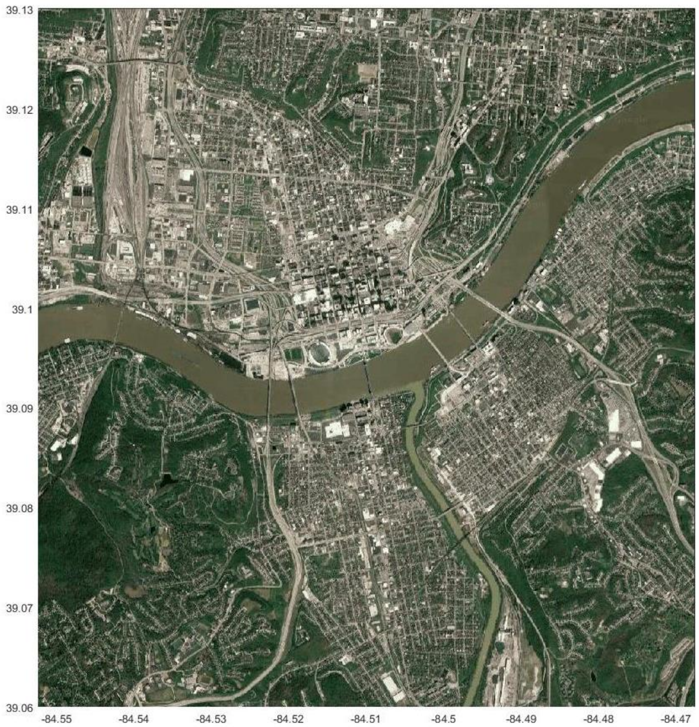


Figure 7: Satellite View of Downtown Cincinnati and Northern Kentucky

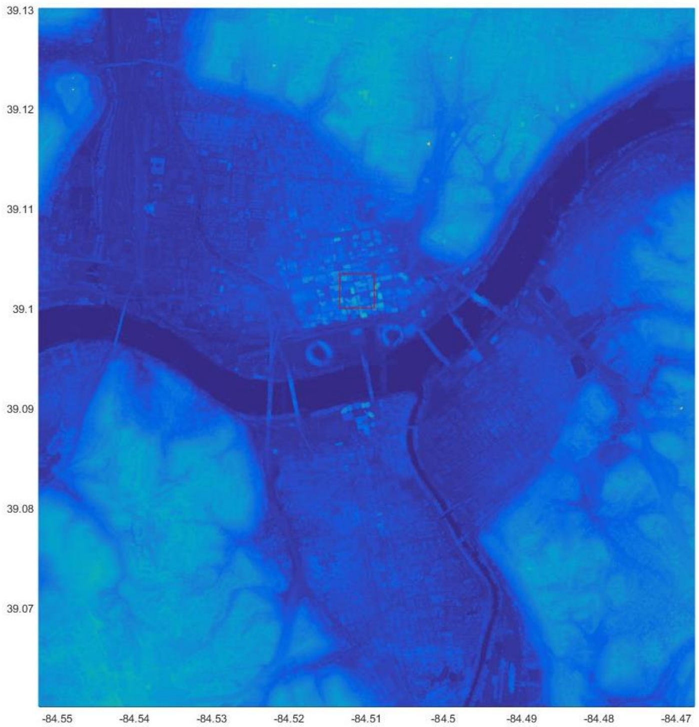


Figure 8: Heatmap of Heights of Downtown Cincinnati and Northern Kentucky

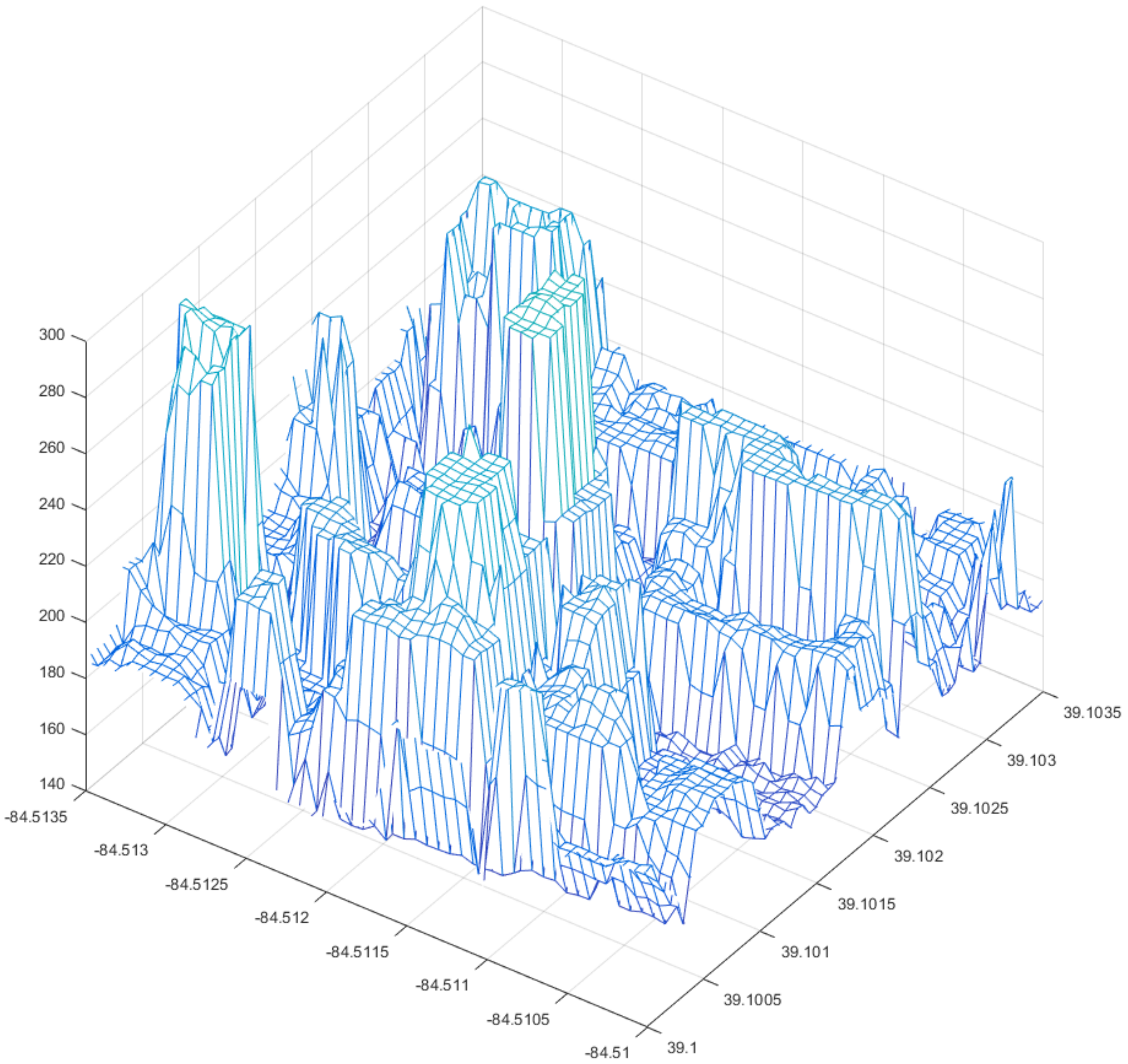


Figure 9: 3-D Representation of the Portion in the Red Box on the Previous Heatmap

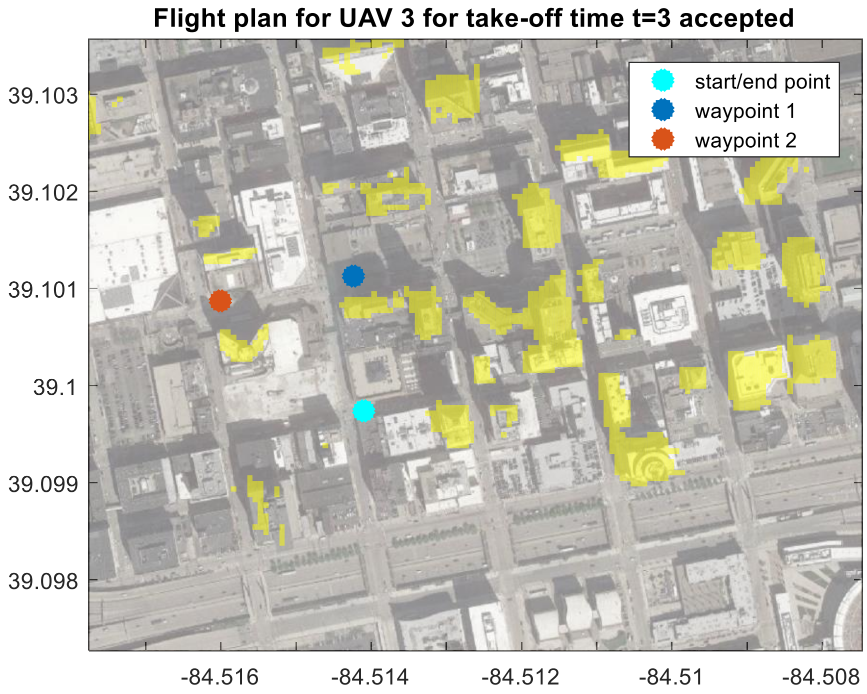


Figure 10: Example of Waypoints given to Algorithm for Path Planning

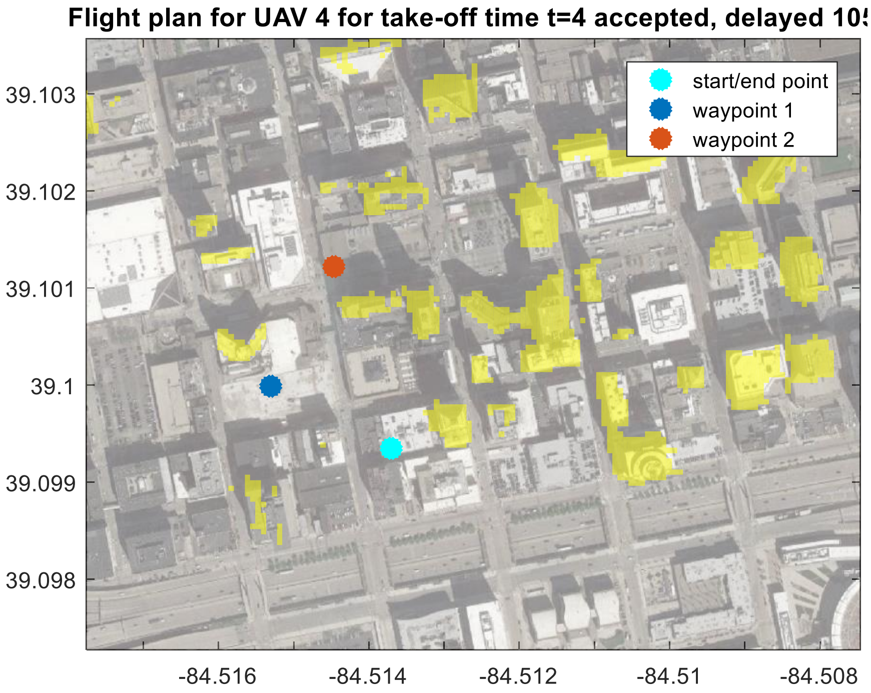


Figure 11: Example of Waypoints given to Algorithm for Path Planning, Delayed because of Existing Flights

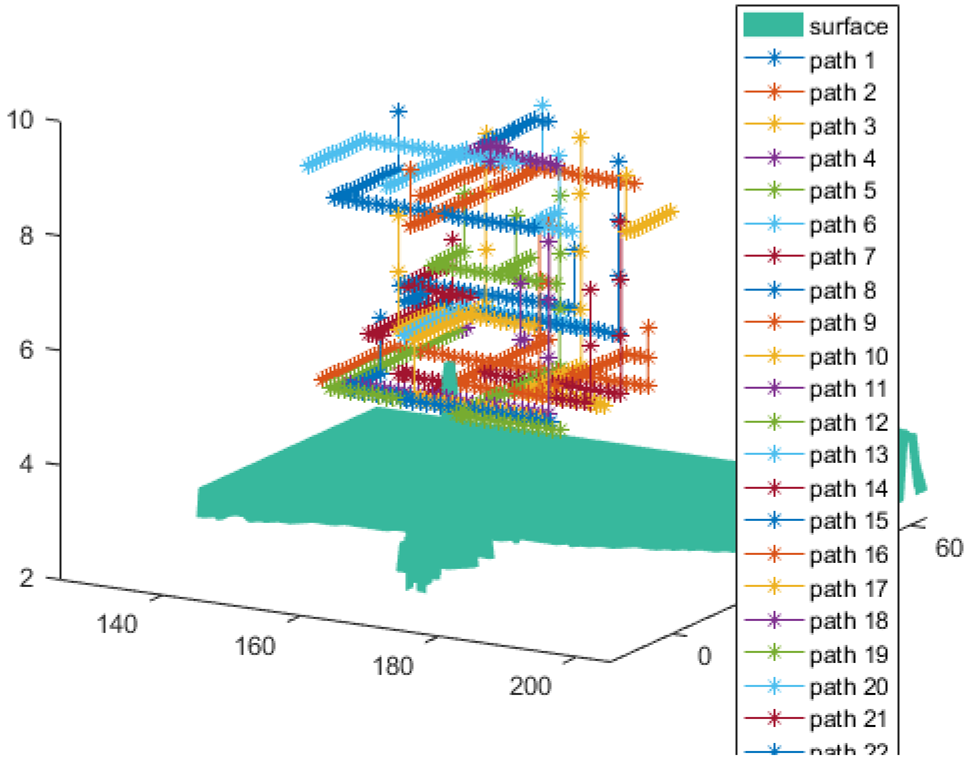


Figure 12: Results of 30 Flights Planned in Close Proximity to Each Other