Optimization Based Path Planning in Dynamic Environments

Matthew Duffield *

Brigham Young University, Provo, UT, 84606, USA

This paper presents a time-based path planning optimizer for separation assurance for Unmanned Aerial Systems (UAS). Given Automatic Dependent Surveillance-Broadcast as a sensor, position, velocity, and identi cation information is available at ranges on the order of 50 nautical miles. Such long-range intruder detection facilitates path planning for separation assurance, but also poses computational and robustness challenges. The time-based path optimizer presented in this paper provides a path planning method that takes advantage of long-range ADS-B information and addresses the associated challenges. It is capable of robust, long-range path planning and is computationally e cient enough to run successively for increased robustness. The ultimate result of this research is a time-based path planner that is suitable for a Detect and Avoid solution on a small UAS in the National Airspace System.

Nomenclature

m	Meters
nmi	Nautical miles
ft	Feet
x	Cartesian x coordinate or direction
y	Cartesian y coordinate or direction
z	Cartesian z coordinate or direction
t	Time in seconds
R	Safety radius
s	Seconds

Subscript

i	Variable number

I. Introduction

A. Motivation

The number of both public and private applications of Unmanned Aerial Systems (UAS) is increasing at an amazing rate. Governmental institutions are taking an interest in UAS for their ability to simply and efficiently perform tasks such as weather research, search and rescue, wildlife surveillance, law enforcement, wildfire monitoring, and military training. The US Department of Transportation has projected that by the year 2035 there will be approximately 70,000 UAS operated by governmental agencies in the US [1]. Private industry is also very interested in UAS applications. Anticipated non-governmental UAS operations include

*Graduate Research Assistant, Department of Mechanical Engineering, Provo, UT, 84606

smoke stack inspection, cinematography, crop dusting, oil exploration, and news and traffic reporting. The demand for UAS operations in the National Airspace System (NAS) is rapidly growing.

The Federal Aviation Administration (FAA) has mandated that for UAS to be permitted in the NAS, UAS must be capable of an Equivalent Level of Safety (ELOS) to the see-and-avoid mandate for manned aircraft [2, 3]. For manned aircraft each pilot has a responsibility to visually scan the surrounding airspace for possible intruding aircraft and take action to avoid a collision. Likewise UAS must be capable of an equivalent degree of monitoring and avoidance of other aircraft. This mandate is known as Detect and Avoid (DAA).

To satisfactorily accomplish the DAA mandate, UAS must be able to both detect other aircraft and plan a collision free path to avoid them. This results in essentially two separate, albeit very related, tasks: detection and avoidance. Many different sensors have been applied to intruder detection for DAA efforts. While radar and visual methods have drawn a particularly large amount of attention [4–7], another promising sensor is Automatic Dependent Surveillance-Broadcast (ADS-B). ADS-B is a cooperative sensor that supports an exchange of position, velocity, and identification information between aircraft at demonstrated ranges of up to 80 nautical miles [8]. In DAA efforts to avoid intruders, such long-range, detailed intruder information is particularly valuable.

Many efforts in collision avoidance focus on small time horizon reactionary avoidance where the goal is to avoid an eminent collision as quickly as possible [9]. The maximum detection ranges for radar and visual methods on small UAS typically lend themselves to this type of approach. However, with the long-range intruder information available through ADS-B, the avoidance paradigm can shift to focus on long-range path planning to avoid the possibility of a collision scenario. This is typically referred to as separation assurance or conflict resolution [10].

In planning a path to maintain separation assurance, the likelihood of two aircraft maintaining a safe distance between them increases. Some of the challenges that accompany long-range separation path planning are the computational expense of long-range path planning, uncertainty in intruder aircraft positions, and unpredictability of intruder aircraft future maneuvers. To develop a path planning method that offers the benefits of ADS-B based separation assurance and to mitigate the challenges associated therewith, the goal of this project is to develop an optimization-based path planner for separation assurance on UAS in a dynamic environment.

B. Relevant Literature

Other research has addressed the problem of optimal path planning for UAS. Sanders and Ray presented an offline path planner for fixed wing UAS using a genetic algorithm [11]. This work successfully demonstrated collision avoidance of static obstacles and incorporated UAS dynamics into the algorithm constraints. A multi-objective approach was formulated to minimize path length and collision threat. The result of this research was a valuable algorithm for static obstacle avoidance, but no further work was reported to extend the research to dynamic obstacles or real-time execution.

Jung, Knutzon, Oliver, and Winer presented a three-dimensional path planning optimizer for UAS using a particle swarm algorithm [12]. This method used a hybrid objective function that had user-defined weights for fuel minimization and threat avoidance. While the work demonstrated avoidance of ground threats, it did not address dynamic aerial threats. Additionally, it relied on an operator to select the final weighting distribution between fuel minimization and threat avoidance.

A linear programming, three-dimensional path planning method for UAS is presented by Chen, Han, and Zhao [13]. This research is particularly applicable to the separation assurance path planning challenge. It presented a linear programming method to plan a path in the presence of dynamic obstacles. The reported execution time is suitable for real-time applications. The overall goal of the algorithm was to find the optimal path along which a UAS could pursue a target and avoid obstacles. This work is very relevant to separation assurance path planning, but the goal and scenario are different. The scenarios demonstrated in the article have distances on the order of 7,000 meters. This is significantly less than the 25,000-100,000 meter range expected in a separation scenario. Ultimately Chen, Han, and Zhao's work could be transformed into a separation assurance path planning method, but further work is necessary to accomplish and demonstrate this.

II. Methodology

The approach for this research is to use gradient-based, constrained optimization techniques to optimize the position of nodes along a path so as to find the minimum length path. The problem formulation, robustness measures, underlying assumptions, and optimizer implementation are considered in this section.

A. Problem Formulation

The overall problem formulation uses a modified Euclidean distance as the objective function, Cartesian coordinates of each node as the design variables, and an ellipsoidal separation criterion as the constraints. This formulation is shown below.

$$\begin{aligned}
\text{Minimize}:&\ \sum_{i=1}^{n-1}(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2\\
\text{W.r.t}:&\ x_1, x_2, \ldots, x_n, z_1, z_2, \ldots, z_n\\
\text{Subject to}:&\ \frac{(x_i - x_{int}(t))^2}{R_{horz}^2} + \frac{(y_i - y_{int}(t))^2}{R_{horz}^2} + \frac{(z_i - z_{int}(t))^2}{R_{vert}^2} > 1
\end{aligned}$$

The Euclidean distance formula shown in Equation 1 provides a very intuitive choice for a path length minimization objective function, but it does pose several inefficiencies.

$$\sum_{i=1}^{n-1}\sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2} \tag{1}$$

Due to the square root operation the function gradient is more complex than necessary. There is a significant improvement in computational efficiency by modifying Equation 1 to be a second order function as shown in Equation 2. While the output of the objective function is not as intuitive, the overall optimization is more efficient.

$$\sum_{i=1}^{n-1}(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2 + (z_i - z_{i+1})^2 \tag{2}$$

The design variables for the optimization are the x and z Cartesian coordinates of each of the path nodes. The y coordinate for each node is excluded to reduce the dimensionality of the problem. This exclusion does not significantly reduce the flexibility of the solution due to the fact that prior to optimization the positions of the start and end node can be transformed to both lie on the y axis. In such an orientation, the most impactful coordinate variation will occur orthogonal to the y axis. Thus the y coordinate is an unnecessary dimension.

Another key consideration in selecting the design variables is the number of nodes between the start and end node. A greater number of nodes adds more degrees of freedom to the solution, but also increases the computational requirements of the optimizer. Fewer nodes requires less computational expense but also reduces the conformability of the path. A brief analysis of the effect of the number of nodes is shown in Section III.

The constraints are designed to ensure that at every time step the ownship maintains separation from each intruder aircraft. As a representation of general FAA separation thresholds of 500 feet vertical separation and 5 nautical miles horizontal separation, the path is constrained to be outside a similarly sized ellipsoidal buffer surrounding each intruder. To do this a set of sub-nodes are created between each path node. The time at which the ownship aircraft will arrive at each sub-node is then calculated. Using that time and a knowledge of the intruder position and velocity, each intruder position is linearly extrapolated into the future. With the propagated ownship and intruder positions, it is possible to ensure that the ownship is outside of the ellipsoidal buffer for each intruder at all times.

B. Robustness

To accommodate for uncertainty in the intruder positions and velocities, feasibility robustness is necessary. The FAA requires ADS-B position and velocity reports to be accurate to certain thresholds [3]. From these thresholds it is possible to derive the error variance for both position and velocity. Such error in the intruder states must be accounted for in any applicable, realistic path planning method. Furthermore the nature of long-range, time-based path planning requires extrapolation of intruder positions over long time horizons. This necessity introduces two forms of error: prediction error and model error. Prediction error results from the growing uncertainty as state information is propagated into the future. Model error also grows as it is predicted into the future, but it results from uncertainty in the model by which the information is propagated.

Two robustness techniques mitigate these two types of error. Feasibility robustness allows for error in constraint parameters to be incorporated into the constraint. In this path planner, a worst-case feasibility robustness is used to account for the uncertainty in intruder position. To implement the feasibility robustness several simplifications are necessary. As error is propagated forward in time, its growth can be calculated by the covariance prediction method outlined in the Kalman filter [9]. Over large time horizons however, this method yields uncertainties that envelop the entire path region including start and end points. This is too large to be useful. As a result, the intruder error is assumed to be constant over the entire time horizon. This constant error value is defined to be six times the standard deviation calculated from the FAA mandated ADS-B accuracy requirements. The six sigma error is then taken as the worst-case deviation. These two simplifications make the path optimization possible and are both accounted for in the second robustness method.

Successive planning robustness is a method by which the path is re-planned at regular intervals. While this is not a new method, it does significantly contribute to the overall applicability of the path planner. Unpredictable maneuvers, environmental factors such as changes in wind, and error prediction simplifications can be addressed by regularly re-planning the optimized path. This is a major reason that computational expense is of interest. For a more rapid path planning methods new plans can be generated more quickly and more often.

C. Assumptions

For this problem formulation several assumptions are necessary. The assumed sensor with which intruder information is gathered is Automatic Dependent Surveillance-Broadcast. This sensor provides latitude, longitude, altitude, ground speed, heading, and climb rate [3]. The intruder positions and velocities used to propagate intruder position into the future are derived from this information. Furthermore the propagation method for intruder positions is a constant velocity method. Thus it is assumed that the intruders are not maneuvering. While this assumption may not be entirely correct, long-range intruder detection, such as is possible with ADS-B, and regular, successive path re-planning can alleviate much of the error in the assumption. The distances used in the intruder ellipsoidal buffer region are FAA mandated aircraft separation distances for aircraft further than 40 nautical miles from an air traffic control radar station.

D. Optimizer Implementation

To solve the path planning optimization problem, we are using MATLAB's fmincon function. The active set method has shown to be the fastest method for this problem. Currently the objective function gradients are calculated analytically and constraint gradients are calculated internally by fmincon's finite difference method. The finite difference method internal to fmincon resulted in faster optimization results than a custom complex step gradient method. The initial starting point path used to seed the optimizer is an adjusted straight line path. To ensure that the initial path is feasible, or at least close to it, all nodes in the path are shifted upward by increasing their altitude by 100 meters. By ensuring that the initial path is feasible, or very close to it, the optimizer is able to converge much more quickly.

III. Testing

The optimization-based path planner for separation assurance on UAS in dynamic environments was tested in simulation to show convergence and effective path planning in both multiple random intruder scenarios and several specific intruder configurations.

A. Simulation

The simulations for testing were executed in MATLAB on an CORE™ i5 processor. The start and end point of the path were placed at (0,0,2500) and (0,27780,2500) meters respectively. This range corresponds to the minimum allowable broadcast range for ADS-B transmissions [14,15]. To ensure separation along the entire path, the constraint was evaluated at 50 sub-nodes between each major node.

An initial intruder scenario for testing during development was devised to include both a crossing and head-on intruder. This scenario was expanded to include three crossing intruders from the left, each at a different altitude, one diagonally crossing intruder from the right, and one head-on intruder. The intruder scenarios presented in this set provided a good starting point for the optimizer testing and will be referred to as Scenario 1. Further testing included scenarios with randomly generated intruders. To create these scenarios, we randomly generated intruder starting positions and velocities, such as would available from ADS-B information. The altitudes of each of the intruders was adjusted to conform to Visual Flight Rules requirements for altitude and heading [3]. This method of intruder generation yielded a wide variety of scenarios for testing.

B. Results

1. Path Results

The primary result of path optimizer is a time-based separation assurance path. Figures 1 and 2 show an optimized path for both eight and twelve nodes respectively. The blue line seen in the plots is the optimized ownship path. Each node is represented by a tick mark on the path. In each figure the left hand plot shows a top view of the path, and the right hand plot shows a side view of the path. Each intruder and the associated separation volume is represented by a shaded ellipsoid. The intruder positions shown are the initial intruder positions. The arrows on the left hand plot, or top view, indicate the direction of each intruder's trajectory.

The optimizer solution is sensitive to the initial path guess. In Figures 1 and 2, the initial guess was the straight line path with the altitude offset by 100 meters. In a different test with the same intruder configuration, the initial path was the straight line path with the altitude offset by 500 meters. This yielded a feasible, but different optimized path. Figure 3 shows the difference in the optimized eight-node path for a 100 meter altitude offset and 500 meter altitude offset starting path.

2. Run Time Results

For UAS applications where computational resources are limited, the run time for the optimizer is of particular interest. Due to the fact that the number of nodes along the path directly influences the number of design variables, the number of nodes significantly effects the run time of the optimizer. Table 1 shows the relationship between run time, path length, and the number of nodes.

The run time listed in the second column of Table 1 is an average of five path optimization runs. Each run started with the 100 meter offset initial path and had a random set of intruders. The third column of Table 1 shows the run time for a single execution of Scenario 1 which is outlined in Subsection A of this section. This scenario has three crossing intruders from the left, one diagonally crossing intruder from the

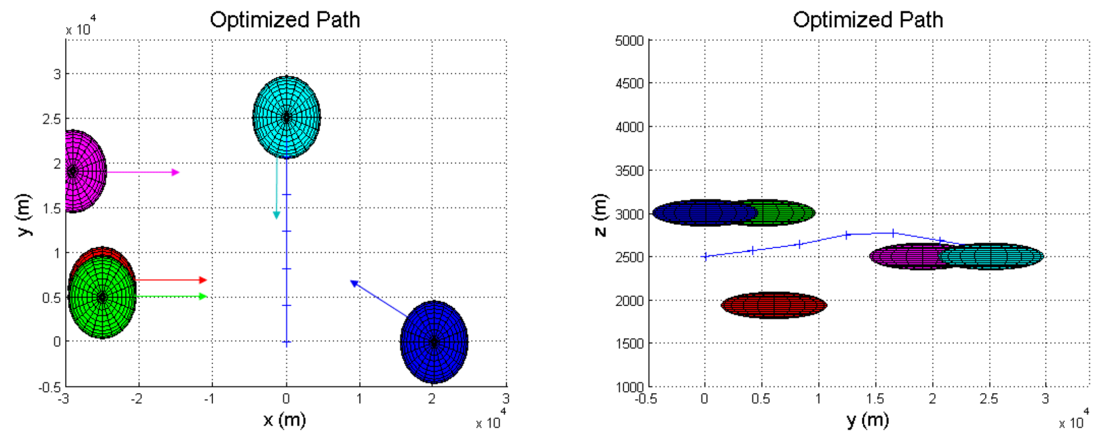

Figure 1. This plot shows results of the preliminary path optimization for eight nodes. The left plot is a top view, and the right plot is aside view.

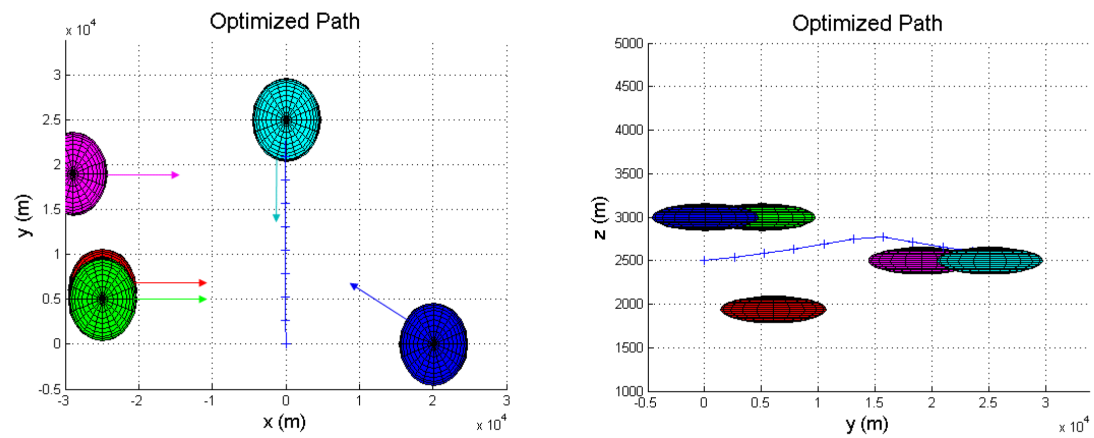

Figure 2. This plot shows results of the preliminary path optimization for twelve nodes. The left plot is a top view, and the right plot is aside view.

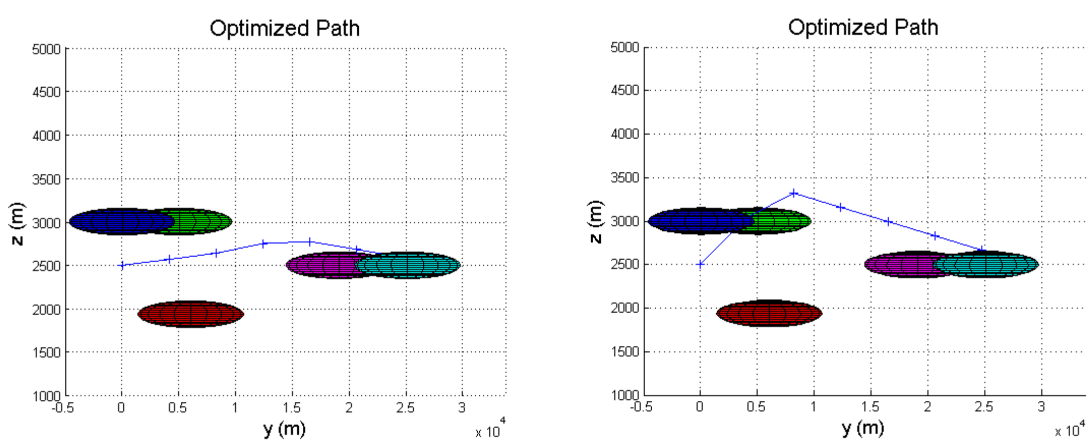

Figure 3. This plot shows two eight-node optimized paths. The left hand plot had a 100 meter altitude offset straight line initial path, and the right hand plot had a 500 meter altitude offset straight line initial path.

Table 1. This table shows the change in run time and path length as a function of the number of nodes in the path including the start and end nodes. The run times listed represent an average run time with random intruder scenarios and the run time for optimization of intruder Scenario 1. The path length reported is the optimized path length for intruder Scenario 1.

Number of Nodes	Average Run Time (s)	Scenario 1 Run Time (s)	Scenario 1 Path Length (m)
6	2.745	2.487	28776.14
8	4.301	5.784	28775.70
10	7.382	9.354	28775.46
12	12.766	12.700	28775.29

right, and one head-on intruder. The fourth column of Table 1 presents the path length associated with the optimization executions in column three. Note that this value is not the objective function optimum. The objective function is a modified Euclidean distance that does not represent actual distance along the path.

3. Robustness Results

Testing of the robustness measures focused on the feasibility robustness. Monte Carlo simulations provided for the testing of the feasibility robustness. Given a uniform distribution of the worst-case deviation about the initial position of the intruders, we simulated 100,000 intruder initial positions. Then the constraints for each sub-node were evaluated for both the non-robust path and worst-case robust path. Each path had six nodes.

Table 2. This table shows the number of conflicts for 100,000 Monte Carlo simulations of intruder positions for Scenario 1. It also shows the difference in path length and run time between the non-robust and worst-case robust paths.

Path Type	Number of Conflicts	Path Lenth (m)	Run Time (s)
Non-Robust	49985	28772.0	1.612
Worst-Case Robust	0	28776.7	3.308

Table 2 shows the results from the Monte Carlo simulation. For the non-robust path, slightly less than half of the simulations resulted in a conflict. For the worst-case robust path, however, there were no conflicts. The path length for both paths is almost identical, but the robust path required more time to run than the non-robust path.

IV. Analysis

The results presented in Section III are promising. Figures 1 and 2 show the results of a path planner that is capable of generating a time-based path through a complex, dynamic intruder environment. The time-based aspect of the path provides several key benefits. In Figures 1 and 2 an intruder is located such that the separation zone totally envelops the end node of the desired path. In a non-time-based path planner, this apparent conflict would require adjustment of the node in order for the planner to find a viable path. With the time-based path optimizer, this is not a concern. The intruder and ownship positions are evaluated for a conflict only at the time at which the two aircraft are at the given position. Thus the time-based path optimizer eliminates many unnecessary maneuvers of non-time-based planners.

The sensitivity of the time-based path optimizer to the initial path guess is shown in Figure 3. This sensitivity suggests that there are local minima in the design space. In finding a safe, conflict free path, the local minima do not inhibit the optimizer, but in finding the shortest path, the local minima result in an inefficiency. For this reason, the initial path is an important consideration. An initial path farther from the unconstrained straight line optimum provides greater assurance that the initial path is feasible, but it increases the likelihood that the optimizer will find a local minimum. On the other hand, a path closer to the unconstrained straight line optimum provides much less opportunity for the optimizer to find a local minimum, but also increases the run time as the optimizer searches for a feasible initial path. Interestingly there is the possibility that in choosing a value very close to the unconstrained optimum, that the optimizer will find an initial feasible guess that is quite far from the true optimum. This may result in the worst case of both scenarios in that the optimizer spends time finding an initial path, and then the initial path that it finds is sufficiently far from the true optimum that there is a high likelihood of the optimizer finding a local minimum. Thus the initial path guess is an important parameter. The choice of the 100 meter altitude offset straight line path, which was used in the majority the results presented, was determine empirically.

Table 1 shows the time necessary to run the time-based path optimizer for different numbers of path nodes. One very interesting correlation in the results is that as the number of nodes increases, the run time increases and the path length decreases. However, the run time increases much more rapidly than the path length decreases. The path length difference between a path with six nodes and one with twelve nodes is only about 0.75 meters. This correlation indicates that when choosing the number of nodes it is reasonable to err on the side of fewer nodes. Another important observation from Table 2 is that all of the run times shown are slower than the 1 Hz measurement rate of ADS-B. While this initially seems to be a serious downfall of the path planning method, it is not a significant drawback. Any implemenation of this path planner on a UAS would require a conversion from MATLAB to C++ or a similar language. In converting from MATLAB to a compiled language, such as C++, the computational expense of the planner will significantly decrease. An additional factor that reduces the impact of slower-than-real-time computation is that since the resulting path of the planner is time-based, it does not lose validity over time in the same way that a non-time-based method does. A time-based path is computed using future positions of both the ownship and intruders. Thus it is theoretically always valid. There is uncertainty in the method by which the intruder positions are propagated forward in time, but this uncertainty is less impactful than the uncertainty associated with considering the intruders to be static. Thus as a result of using an interpreted language for testing, and the increased validity period associated with a time-based path, the seemingly slow run times are not a significant concern.

The importance of robustness is illustrated by Table 2. From this table it is clear that the addition of worst-case robustness drastically improves the feasibility of the optimized path while only adding 3.7 meters to the overall path length and requiring an extra 1.7 seconds of computation time. This increase in robustness significantly improves the applicability of the optimized path and makes it useful in a realistic environment with uncertainty in intruder positions.

V. Conclusions

In conclusion, the time-based path optimizer presented in this paper is capable of long distance path planning for separation assurance in an environment with dynamic obstacles. Evaluation for separation assurance at multiple sub-nodes between the primary path nodes allows for high resolution long-range planning without excessive computational cost. The incorporated robustness measures result in a path that is viable in the presence of uncertainty in intruder positions. Ultimately this time-based path optimizer is a capable long-range path planner and is a key step toward a Detect and Avoid solution for UAS in the National Airspace System.

References

[1] "Unmanned Aircraft System (UAS) Service Demand 2015-2035," Tech. rep., U.S. Department of Transportation, 2013.

[2] Hottman, S., Hansen, K., and Berry, M., "Literature Review on Detect, Sense, and Avoid Technology for Unmanned Aircraft Systems," Tech. Rep. September, 2009.

[3] Federal Aviation Administration, "SUBCHAPTER F — AIR TRAFFIC AND GENERAL OPERATING RULES PART 91 — GENERAL OPERATING," 2012, pp. 579–844.

[4] Mackie, J., Spencer, J., and Warnick, K., "Compact FMCW Radar for GPS-Denied Navigation and Sense and Avoid,"

IEEE Antennas and Propagation Society, AP-S International Symposium, 2014, pp. 989–990.

[5]Dey, D., Geyer, C., Singh, S., and Digioia, M., "Passive , long-range detection of Aircraft : Towards a field deployable Sense and Avoid System," *Field & Service Robotics*, 2009.

[6]Lai, J., Ford, J. J., Mejias, L., Shea, P. O., and Walker, R., "See and Avoid Using Onboard Computer Vision," *Sense and Avoid in UAS: Research and Applications*, chap. 10. See an, 2012, pp. 265–294.

[7]Mejias, L., McNamara, S., Lai, J., and Ford, J., "Vision-based detection and tracking of aerial targets for UAV collision avoidance," *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010, pp. 87–92.

[8]Moody, C. and Strain, R., "Implementation Consideration for Automatic Dependent Surveillance - Broadcast on Unmanned Aircraft Systems," *AIAA Infotech@Aerospace Conference*, American Institute of Aeronautics and Astronautics, Reston, Virigina, April 2009, pp. 1–8.

[9]Sahawneh, L. R., Beard, R. W., Avadhanam, S., and Bai, H., "Chain-based Collision Avoidance for UAS Sense-and-Avoid Systems," *Aerospace Robotics and Unmanned/Autonomous Systems IV*, 2013, pp. 1–15.

[10]George, S., "Concepts of use for UAS Sense and Avoid Equipment," Tech. rep., 2009.

[11]Sanders, G. and Ray, T., "Optimal Offline Path Planning of a Fixed Wing Unmanned Aerial Vehicle (UAV) using an Evolutionary Algorithm," *Evolutionary Computation*, 2007, pp. 4410–4416.

[12]Foo, J. L., Knutzon, J. S., Oliver, J. H., and Winer, E. H., "Three-dimensional path planning of unmanned aerial vehicles using particle swarm optimization," *... Analysis and Optimization ...*, No. September, 2006, pp. 1–10.

[13]Chen, Y., Han, J., and Zhao, X., "Three-dimensional path planning for unmanned aerial vehicle based on linear programming," *Robotica*, Vol. 30, No. 05, 2012, pp. 773–781.

[14]Administration, F. A., "Automatic Dependent Surveillance— Broadcast (ADS–B) Out Performance Requirements To Support Air Traffic Control (ATC) Service," 2010.

[15]Helfrick, A., *Principles of Avionics*, Avionics Communications Inc., eighth ed., 2013.