

A fuzzy logic approach to collision avoidance in smart UAVs

Technical Report #CSSE12-05

Michelle Hromatka
Computer Science Dept.
College of Saint Benedict | Saint John's University
St. Joseph, MN
mthromatka@csbsju.edu

Jeffrey West
Mechanical Engineering Dept.
University of Southern California
Los Angeles, California
jeffrey.west@usc.edu

Matt Holt
Computer Science Dept.
Auburn University
Auburn, AL
holtjma@tigermail.auburn.edu

Saad Biaz
Computer Science Dept.
Auburn University
Auburn, AL
biazsaa@auburn.email.edu

Abstract—In recent years there has been an increase of interest related to collision avoidance techniques for unmanned aerial vehicles. Many approaches have been proposed but very few have been tested in high density, fixed speed test conditions. Our approach uses fuzzy logic to determine an appropriate avoidance maneuver after a possible collision is detected. Initially fuzzy logic was also used to detect possible collisions, the reasons this level was removed from the final results are detailed within. This method was chosen for its adaptability, ease of implementation, and robustness with inadequate sensing techniques. This approach is explained in detail and the simulated results are presented after testing on two different field sizes, each with four levels of plane density.

Index Terms—UAV; collision avoidance; fuzzy logic;

I. INTRODUCTION

Collision avoidance techniques for unmanned aerial vehicles have been a popular area of study in the field of intelligent robotics. In order for these vehicles to be used in areas such as surveillance or exploration, the vehicles must be able to autonomously react to static or dynamic obstacles while maintaining an efficient path. Many approaches have been taken, including path planning, potential fields, geometric and fuzzy logic methods. We first outline the problem in section two, then move on to describe the various approaches to solving the problem, including our chosen approach of using fuzzy logic to avoid detected collisions.

II. PROBLEM STATEMENT

Many algorithms exist to solve the general problem of UAV collision avoidance, but few of these algorithms have been tested in similar test conditions. In order to synthesize results and compare many algorithms in a single simulation, Holt et al. developed a simulator for comparison of various algorithms

[1]. By using the simulator, different algorithms can be tested using the same metrics and flight constraints on the same platform. All algorithms are tested in two dimensions, with UAVs traveling at a constant speed of 25 miles per hour. Turning radius was modeled after an EasyStar RC aircraft and limited to 22.5 degrees per second. Telemetry updates are retrieved once per second, and new waypoints directing the UAV can be sent once per second. Environmental factors are also excluded from simulation.

The simulator was designed to test algorithms for real-time calculations. Regardless of how long calculations take, the simulator runs in real-time and new telemetry updates will be generated once per second. If no new waypoints are generated from the calculations in time, the current heading will be followed until a waypoint is received.

Two metrics were chosen to determine effectiveness. A *near miss* is defined as one UAV traveling within a radius of 12 meters of another UAV. This is meant to represent a collision, as 12 meters represents approximately the distance travelled in one timestep, and should be avoided at all costs. Similarly, a *conflict* is defined as one UAV traveling within a radius of 24 meters of another UAV, or approximately two timesteps. This represents the danger zone and should be avoided, but is not as critical as a near miss. Both are shown in Figure 1.

A separate metric was chosen to provide a measure of effectiveness. This is determined by the number of goal positions, or waypoints that a UAV achieves. An overly cautious algorithm may avoid near misses or conflicts very well, but the overall number of waypoints reached will be decreased. A delicate balance of effectiveness and efficiency for each algorithm must be achieved. The simulation time limit is ten minutes, and based on the number of near misses

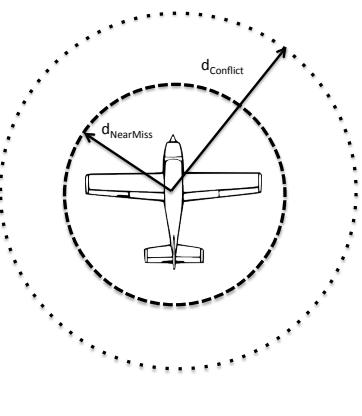


Fig. 1. Near Miss and Conflict

and waypoints reached, an algorithm's effectiveness can be measured. Note: for the purposes of this paper, parameters were optimized for low collisions (high effectiveness). As with any algorithm, parameters can be adjusted for better efficiency if some collisions are an acceptable sacrifice for better efficiency.

Previous research has tested Mixed Integer Linear Programming (MILP), Dynamic Sparse A* Algorithm, and Artificial Potential Fields in this simulation [1]. Each algorithm was run in eight different field situations: a 500m square field and a 1000m square field, each tested with 4, 8, 16 and 32 planes. This will allow for easy comparison of methods presented here to those previously tested.

III. LITERATURE REVIEW

The literature review is split into the following sections: path planning, potential fields, proportional navigation, fuzzy logic, and hybrid approaches. Within these sections, the previously tested Mixed Integer Linear Programming (MILP), Artificial Potential Fields (APF), A* Grid Based Algorithm will be reviewed and explored for strengths and weaknesses. As mentioned in [1], the general goal of this research is to compare various algorithms using the same simulation and metrics, as outlined above. The next section will provide a comprehensive literature review of fuzzy logic as it has been used in collision avoidance. The justification of this approach will be outlined in the methods section.

A. Path Planning

Path planning methods have also been researched using different techniques for finding an optimal path such as Mixed Integer Linear Programming (MILP) and A* search. This approach tries to plan a collision free path for each plane in the field space. There are some other path planning techniques, but these are mainly used to increase efficiency in the chosen path, not for finding a collision-free path.

1) MILP: [2] successfully used MILP for a solution to single and multi-vehicle path planning algorithm with both static and moving obstacles. One of the major advantages of MILP solvers is the ability to limit inputs and outputs; this is useful in UAV collision avoidance where the maximum turning radius is clearly defined. A receding horizon is a common implementation of the MILP approach. This strategy limits the solution to a specified number of time steps into the future, as opposed to solving the entire system, thus an optimal path is found multiple times. A fixed arrival time approach finds an optimal path once but for the entire field. This method is often limited by heavy computational load; the solver is not guaranteed to come up with a solution, especially for many UAVs in a relatively small space, before collisions occur [2]. For this reason the receding horizon approach is widely used, but still not infallible. A horizon time limit should not be so short that the vehicles will not have sufficient time to react and avoid collisions, nor should the time be so long that heavy computational load is still an issue. There are some optimization techniques that have been used to counter the issues that come with truncating the horizon time [3]. According to the simulation from Holt et. al., MILP was tested using a licensed online MILP solver in high density testing conditions. The receding horizon approach and other optimization techniques were used, however, the system still failed for more than 8 vehicles in a 500m square airspace [4]. The optimization techniques used were solver time limit, specifying the max amount of time before reverting to a sub-optimal solution, and the use of a "loiter cycle" which the UAV enters until the solver can come up with a solution. The latter solution is not ideal when evaluating an algorithm's efficiency.

2) A Search Algorithm:* Another path planning algorithm is that of A* search, or the *branch and bound* method. In this technique, the field space is divided into a grid and only adjacent squares are considered as a next point along the path. This consideration of other nodes is called *branching*, estimating the best path from the current square is known as *bounding*. When a square is branched, the cost is computed based off of a heuristic cost function. Each square with its corresponding cost is inserted into a tree structure, typically a heap, which stores the minimum element as the root. As the heap is updated from node to node, the lowest cost path is always at the top of the heap. If the path that the algorithm is considering is no longer the lowest cost, a lower cost option is readily available, and that path is followed until either a lower cost path is found or the goal point is reached [5]. One of the major issues with the A* search approach is determining an appropriate cost function. As with other path planning techniques, computation time is an issue for real-time collision avoidance. There are some ways to lower computation time, especially if the optimum solution is not required but a nearly optimal path is acceptable.

Other literature has used the A* search to determine the best path for a humanoid robot to place its footsteps in an obstacle littered space [6]. In this case, the foot placement options were limited to 15 locations from each previous footstep.

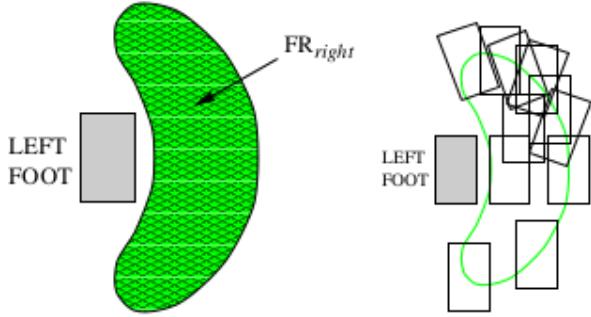


Fig. 2. The degrees of freedom allowed for the humanoid robot's right foot [6]

For their path, specified at 18 footsteps, the unconstrained A* search tree would contain approximately 10^{21} nodes. By limiting the options that A* considers, the final tree contained around 830,000 nodes, significantly less than the predicted 10^{21} nodes. By limiting the degrees of freedom allowed in the bounding step, the computation time is decreased dramatically, however, this approach is still too computationally demanding for high density systems. As seen in [7] and [5], this algorithm was tested in situations of varying plane density using the maximum turning radius as a limitation on the degrees of freedom allowed. In each of these cases, A* performed well for low density airspaces, but still yielded many near-misses for dense systems.

B. Potential Fields

1) *Artificial Potential Fields*: The potential field path planning approach can be split into two different categories: global and local. The global solution requires a priori knowledge of the field and location of the obstacles within that field. Negative charges are applied to the obstacles and the goal point is assigned a positive charge. The robot is then driven to the goal point by the forces equal to the negative gradient of the field [8].

Similar to other global path planning algorithms, global potential field path planning has very high computation costs, especially with robots that have many degrees of freedom, which makes it unsuitable for real-time path planning. The local potential field approach is much more suitable in real-time solution planning. Real-time, local solutions allow for both static and dynamic obstacles, making solutions much more reactive and realistic. The local potential field approach is set up similarly to the global approach, the path for the robot determined by the sum of forces acting on the robot [10]. Potential field path planning methods are computationally efficient for finding a local obstacle-free path, however this method is not without its faults. As [8] clearly asserts, there is a well-documented problem with the potential field approach: local minima. When a robot experiences equal and opposite forces acting upon it, the robot can become stuck in a local minimum. In purely dynamic obstacle avoidance environments, the local minimum problem is likely to have

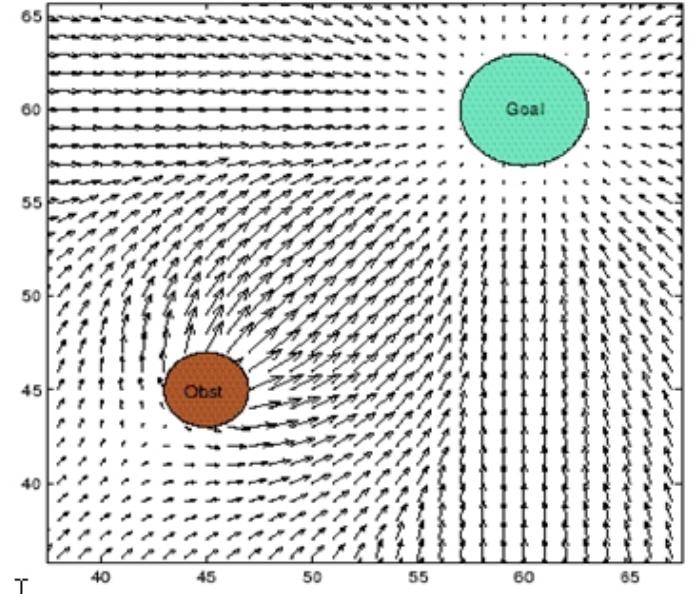


Fig. 3. The gradient along which the UAV is directed [9].

resolved itself by the next timestep, however the interval between timesteps might be enough that a possible collision becomes unavoidable.

2) *Total Field*: Several implementations of UAV collision avoidance using real world applications have also been used in addition to simulation studies. One approach uses magnetic fields and magnetic sensors to detect nearby UAVs [11]. The goal of this approach is to have decentralized collision avoidance. A gradient is computed based on objects sensed, and the UAV will be directed to travel in opposition to the gradient. Field size decreases by the inverse of distance cubed. The results were found through simulation, but some real world magnetic sensor readings and reaction tests were conducted to see if the movements seen in the simulator were realistic. These methods worked well for collision avoidance, but yielded sub optimal paths.

For the purposes of this paper, any algorithms are tested in simulation, but the research goal of developing a decentralized method of collision avoidance is seen as advantageous. As Sigurd states, the problem of collision avoidance rapidly becomes intractable using centralized methods as the number of vehicles increases, making a decentralized approach where each vehicle is responsible for its own trajectory planning desirable [11].

C. Proportional Navigation

Through the course of literature review, several other types or variations of algorithms were considered. One such algorithm, Proportional Navigation-Based Optimal Collision Avoidance Guidance for UAVs is widely used in missile guidance problems, and it can be used in collision avoidance by guiding the relative velocity vector to the collision avoidance vector [12]. The research proposed a collision avoidance law based on conventional Proportional Navigation (PN) guidance

law. First, a collision avoidance vector is defined and the heading of the aircraft is directed toward this vector. The total approach is called Proportional Navigation-based Collision Avoidance Guidance (PNCAG). The approach uses the collision cone geometric method to determine when to enter the collision avoidance mode [13]. Several conditions for collision avoidance must be met including a range between aircraft and obstacle greater than safety distance, the direction of relative velocity must be outside of the collision cone, and the obstacle must be located behind the direction of the relative velocity vector. These techniques have shown promising results in both efficiency and effectiveness.

D. Fuzzy Logic

A common technique used in ground robot collision avoidance is that of fuzzy logic. Fuzzy logic is a form of many valued logic which instead of outputting a 0 or 1, true or false value, instead outputs a degree of truth. Fuzzy membership sets will map the crisp value to a fuzzy value. For example, if height:short is defined as 50 - 70 inches, and height:tall is defined as 65 - 85 inches, a person could belong to both sets height:short and height:tall because fuzzy logic allows values to be a member of two sets at once [14]. This is a more intuitive form of logic and a better representation of the way humans think. A person is not always exclusively short or exclusively tall, rather someone is "kind of tall". Similarly, fuzzy logic can be applied to collision avoidance with output membership sets such as "very left" or "slightly right." Using a fuzzy logic system also allows for sensor fusion, when data from multiple sensors are combined.

For these reasons, fuzzy logic research is frequently used and shown to work in mobile ground robots where sensing techniques are generally inexpensive and inaccurate. Sensor fusion has been accomplished using fuzzy logic in a sense-act approach based on multiple sensors distributed around a 2-D ground robot [15]. This approach was used in situations containing only one robot and stationary obstacles. The mobile ground robot was run through a path containing obstacles where the smoothness of the path was evaluated. Advantages of the fuzzy logic approach are intuitive linguistic terms and a smaller computational load than APF and many other approaches. Fuzzy logic is also useful for unknown and semi-unstructured environments. Sense-act is uninterrupted motion with short development time and incorporates flexibility, easy to adjust parameters.

E. Hybrid Approach

1) Fuzzy Logic / Electrostatic Potential Fields

Many approaches to UAV collision avoidance use a combination of two or more methods. An example of one of these hybrid approaches used fuzzy logic to control an autonomous mobile robot(AMR) while travelling along a pre-planned path generated by a total field path planning technique. A 2-layer fuzzy logic (FL) interface was combined with an electrostatic potential field path planner (EPF) to provide real-time mobile robot navigation in a 2-D dynamic environment [16]. The

first layer of fuzzy logic provided a sensor fusion system to combine data from several sonar sensors returning two fuzzy variables, direction and distance in four directions: front, back, left, and right. The second layer provided an obstacle detection module that outputs the position and degree of possibility which a collision may occur. This information is combined in motion control module with the path planner output. If an obstacle blocks the pre-planned EPF path, collision avoidance maneuvers affecting heading and speed are used to avoid the detected collision.

This EPF/FL approach combined planned and reactive behavior into one algorithm. EPF path planning reacts slowly to unknown moving obstacles. The EPF/FL approach allowed EPF to plan the path and allowed the fuzzy inference system to implement the path while avoiding collisions with all obstacles. Some discussion was given into tuning the membership functions, however, no analytical method guarantees optimal selection for the Mamdani-type FL inference systems. Several approaches have been used to tune membership functions, such as neural nets, genetic algorithms, and neuro-fuzzy methods, but none of these approaches guarantee an optimal solution [16]. Additionally, advantages of EPF compare to other potential field solutions were concluded. Collision free paths generated by the EPF method necessarily lead the UAV toward the goal, local minima are not generated within the field and stagnation points within the field do not exist. The path planned by EPF also generates an optimal minimum occupancy path toward the goal [16].

There are several limitations when applying much of this research to UAV collision avoidance. First, many of the obstacles in this research are static, which is not directly applicable to UAV collision avoidance with multiple UAVs. Mobile ground robots have fewer limitations of freedom of motion than do UAVs. Changes in speed, reversing direction and a 360° turning radius are components of many 2-D ground robots, all three of which are not allowed or limited in this UAV collision avoidance research.

One example of using fuzzy logic in UAV collision avoidance was found. In this scenario, a pre-planned path for the UAV was used until a collision was detected. The obstacles were of varying shape and size and the avoidance technique sensed these two factors, then found an avoidance maneuver that deviated from the pre-planned path. Again, obstacles were static and changes in speed were allowed in determining an appropriate avoidance maneuver. The UAV was able to successfully avoid collisions with one or multiple obstacles in all cases presented [17]. More information on this approach is detailed in the Methods section, as we use this as the basis for the formation of our fuzzy logic collision avoidance system.

2) Path Planning / Reactive Avoidance

Another hybrid approach used a hybrid of path planning and reactive collision avoidance [18]. This algorithm calculated a Dubins path for each plane then considered only the most imminent predicted collision for each aircraft. The line of sight (LOS) rate between the two planes is then increased. When a UAV detects multiple planes that are on a collision

course with itself, it will select the most imminent collision to avoid that has a predicted zero effort miss (ZEM) of less than the minimum desired separation. The LOS is increased by applying a lateral acceleration to increase the rate of rotation of the LOS connecting these two UAVs. The UAV takes the sharpest turn available until the collision course is avoided.

This research uses very different simulation and metrics to test this algorithm. Planes initial positions are chosen to be evenly spaced around a circle that dictates the outside of the test space. A random goal point is published for each plane, somewhere on a smaller concentric circle. This setup eliminates the chance that random plane positions will be chosen in the test space that are actually *inside* or very near to a conflict zone of another plane. The PNCAG's simulation setup, however, does not take into account multiple waypoints. Efficiency of the algorithm was measured by deviation from initial Dubins path.

PNCAG was found to be very efficient as well as effective in reducing the number of near-misses. With an outer circle radius of 500 meters, 20 to 60 UAVs were simulated, giving comparable scenarios to current research goals. The efficiencies achieved were very good (around 90 percent or higher), however, the research cannot be completely compared to results here because multiple waypoints were not achieved (the values were averaged over 10 runs with a single waypoint goal here) and the simulation starting points and goals were chosen in an advantageous manner, as described before.

3) Passive Pre-Planning / Geometric Reactive

Another hybrid approach used a passive approach to collision detection and avoidance [19]. The research split the collision avoidance process into three steps, all defined by two plane's relative bearing angle: detection, avoidance, and an optimal return to pre-planned path. After calculating the bearings between the two planes that are at risk of colliding, a geometric approach is used to determine the necessary turning angle necessary to avoid a collision. The minimum separation distance between two UAVs can also be specified. Results of this method were promising, but only two scenarios, overtaking and front collision were presented in this research. This research is more applicable as an addition to algorithms that may have limited effectiveness in these two special cases.

IV. METHODS

Methods here build upon research described previously: work done by Dong et al. and Tsourveloudis. The two-tiered fuzzy logic system used in Tsourveloudis [16] was combined here with the fuzzy logic collision avoidance used by Dong [17] to develop an efficient and effective algorithm that could handle high-density flight situations.

A. Collision Detection System

Similar to Tsourveloudis' multi level fuzzy system, the first level fuzzy logic controller served as a method to detect possible collisions. Considering only the closest UAV, the first tier used fuzzy logic to output a danger value denoting the likelihood of a collision between the current UAV and the

closest UAV. The danger value was determined by inputs of *distance to collision*, (d_{col}), and *overlap distance*, (d_{ovlp}). Both variables are defined below and a graphical representation of these two inputs can be seen in Figure 4.

$$d_{col} = \min(d_A, d_B) \quad (1)$$

$$d_{ovlp} = d_A - d_B \quad (2)$$

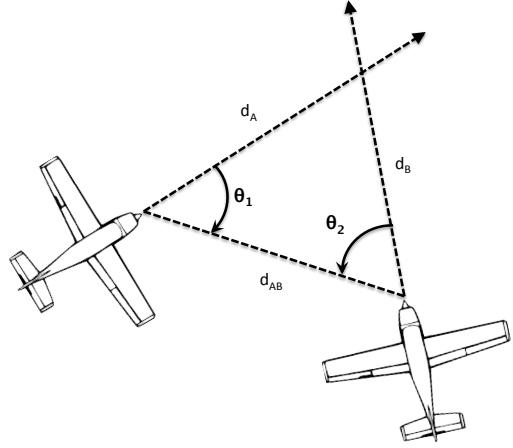


Fig. 4. Common Variables

Distance to collision represents an urgency component in detecting a collision. In low density situation, the closest UAV may be many hundreds of meters away from the current UAV in which case either one or both of the UAVs may have changed course prior to reaching the determined collision point. Overlap distance represents a timing factor; if one UAV will reach the collision point outside of the minimum separation distance between two planes, a collision will not occur despite the fact that the paths of two UAVs will intersect. Overlap distance will preserve the efficiency of the algorithm and only avoid a collision point if a collision will actually occur.

Only the closest UAV is considered in each time step. Using the closest UAV is a simplification, but several improvements upon this technique are discussed in the Future Work section. In some cases, distance to collision and overlap distance will not detect a possible collision. For example, two planes flying parallel could be inside each other's *near miss* zone, but the planes do not have intersecting paths and the danger value returned by the fuzzy logic would incorrectly be low. This is also discussed in the Future Work along with proposed changes. For these two reasons, the first tier fuzzy engine was left out of the final results, in favor of a simple if statement determining whether the closest plane is within a certain danger radius of the current UAV. The danger radius chosen here was 100 meters. Improvement of these techniques is beyond the scope of this research, but highly recommended for future work as including this level of collision detection has the potential for vast improvements on efficiency.

B. Collision Avoidance System

The second tier used a fuzzy logic engine to determine a change in heading for the UAV. This tier was originally only to be entered for high danger values outputted from the first fuzzy system. As stated above this was changed so that this level is entered only if the closest UAV is within a danger radius surrounding the current UAV. Many variations of the second tier fuzzy engine were designed and implemented. Inputs of distance between planes, bearing angle, output from the first fuzzy system, and distance to next waypoint were all considered as possible inputs.

Dong et al. explored fuzzy logic collision avoidance techniques between UAVs and static obstacles. This work is meant to extend the fuzzy logic approach used by Dong et al. to also be applicable to dynamic collision avoidance. As a frame of reference, that approach was duplicated here and results can be found below. Dong's approach, while very useful for static obstacle avoidance, was not found to be an effective approach for dynamic collision avoidance between a group of UAVs in limited airspace.

As a matter of implementation, several changes were made to Dong's methods. First, the simulator used here does not allow for a change in speed, so no fuzzy output for speed was considered. Second, Dong's approach limited the turning radius to $\pm 45^\circ$ but the simulator here sets the limit to $\pm 22.5^\circ$.

Several techniques were used to improve the effectiveness of fuzzy logic in dynamic collision avoidance. In order to eliminate oscillations in many cases, bearing of the opposite plane was added to the general approach described by Dong et al. The fuzzy sets for bearing angles and distances vary slightly as well. The output of change in heading is limited to five fuzzy set possibilities, simplifying rule choices. The final design of the fuzzy logic approach is outlined below. Figures 5, 6, and 7 show the ranges and implementations for each fuzzy input and output set. The bearing angle set is identical for the bearing angle of the current plane, θ_1 , and that of the opposite plane, θ_2 .

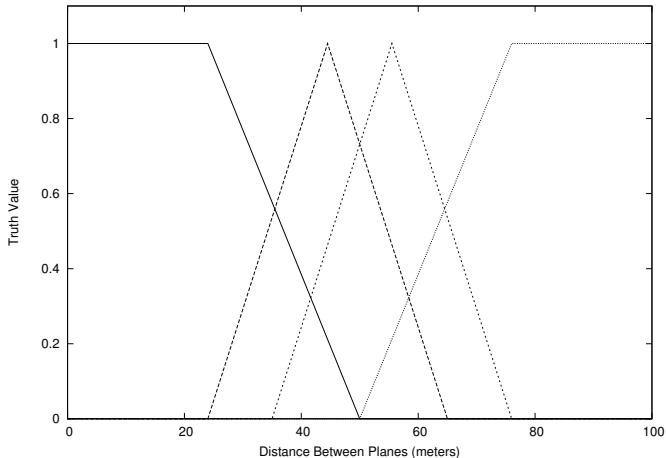


Fig. 5. Distance Fuzzy Sets

The linguistic terms used to describe and their respective

abbreviations from left to right are as follows: *Very Close* [VC], *Close* [C], *Far* [F], *Very Far* [VF].

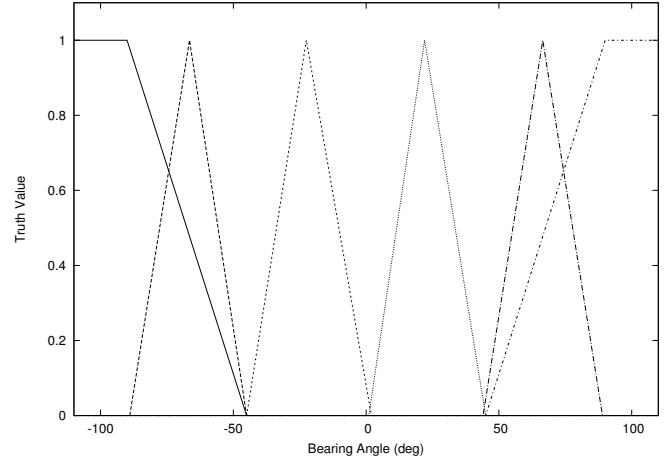


Fig. 6. Bearing Angle Fuzzy Sets

The linguistic terms used to describe and their respective abbreviations from left to right are as follows: *Very Negative* [VN], *Negative* [N], *Low Negative* [LN], *Low Positive* [LP], *Positive* [P], *Very Positive* [VP].

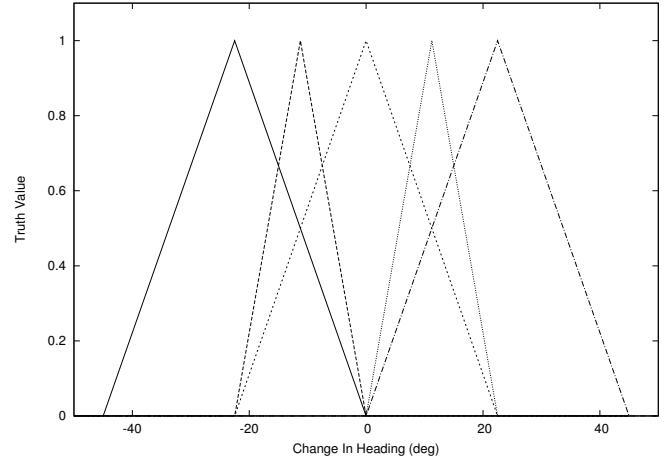


Fig. 7. Change In Heading Fuzzy Sets

The linguistic terms used to describe and their respective abbreviations from left to right are as follows: *Very Left* [VL], *Left* [L], *No Change* [NC], *Right* [R], *Very Right* [VR].

The range of values were chosen based on the limitations of the aircraft in the simulator. Turns are limited to $\pm 22.5^\circ$, corresponding to the center of the outer fuzzy sets, ensuring that the fuzzy engine never returns a change in heading greater than the maximum value allowed.

The rules used for the final testing in simulation are outlined in the following tables. Tables I - IV correspond to the fuzzy sets of VC, C, F, VF for Distance Between Planes, respectively. θ_1 corresponds to the fuzzy sets for the current plane's bearing angle. θ_2 corresponds to the fuzzy sets for the opposite plane's bearing angle, or in other words the plane under consideration for a possible collision.

TABLE I
FUZZY ENGINE RULE SET - PLANE DISTANCE VERY CLOSE [VC]

		θ_2				
θ_1	VN	N	LN	LP	P	VP
VN	R	R	R	R	VR	VR
N	R	R	VR	VR	VR	VR
LN	VR	VR	VR	VR	VL	VL
LP	VR	VR	VR	VL	VL	VL
P	VL	VL	VL	VL	VL	L
VP	VL	VL	VL	VL	L	L

TABLE II
FUZZY ENGINE RULE SET - PLANE DISTANCE CLOSE [C]

		θ_2				
θ_1	VN	N	LN	LP	P	VP
VN	NC	NC	R	R	R	R
N	NC	R	VR	VR	VR	VR
LN	VR	VR	VR	VR	VL	VL
LP	VR	VR	VR	VL	VL	VL
P	VL	VL	VL	VL	L	NC
VP	L	L	L	L	NC	NC

TABLE III
FUZZY ENGINE RULE SET - PLANE DISTANCE FAR [F]

		θ_2				
θ_1	VN	N	LN	LP	P	VP
VN	NC	NC	R	R	R	R
N	NC	R	VR	VR	VR	VR
LN	VR	VR	VR	VR	VL	VL
LP	VR	VR	VR	VL	VL	VL
P	VL	VL	VL	VL	L	NC
VP	L	L	L	L	NC	NC

TABLE IV
FUZZY ENGINE RULE SET - PLANE DISTANCE VERY FAR [VF]

		θ_2				
θ_1	VN	N	LN	LP	P	VP
VN	NC	NC	R	R	R	R
N	NC	R	VR	VR	VR	VR
LN	VR	VR	VR	VR	VL	VL
LP	VR	VR	VR	VL	VL	VL
P	VL	VL	VL	VL	L	NC
VP	L	L	L	L	NC	NC

V. RESULTS

All results were simulated using the ROS based simulator system as described in [1]. In the simulator, a random course file is generated which includes 50 waypoints for each plane, including random start points. For the field sizes of 500m square and 1000m square, each with 4, 8, 16 and 32 planes, three random course files were generated. A note on the simulator: although a plane may experienced a near-miss, the plane continues to fly throughout the simulation. This is to keep the flight situation the same throughout the simulation run time. If UAVs left the simulation after experiencing a near-miss, the simulator would favor an algorithm that stabilized quicker than others. A collision avoidance algorithm may operate very well with 24 UAVs in a 500m airspace, so as soon as enough planes have near-misses, the simulation would stabilize. The purpose of this research is not to find

the breaking point of an avoidance algorithm, but to test how well an avoidance algorithms can handle high-density flight situations.

Each course was run in the simulator without collision avoidance for every scenario as frame of reference. Results of Dong's methods and the methods presented here are compared to this base case. Tables V and VI display the base case results.

As mentioned previously, Dong et al made great progress on collision avoidance of UAVs with static obstacles. Their methods are repeated here for collision avoidance between UAVs in limited airspace. Results of Dong's methods applied to dynamic collision avoidance are displayed below in Tables VII and VIII.

This approach improved upon the base case in most categories and field sizes, but can be optimized further. Changes mentioned previously, such as adding the bearing angle of the opposite plane as a fuzzy input, were made to the methods of Dong et al and the results are detailed below. Results from methods used in this research are displayed in Tables IX and X.

TABLE V
NO COLLISION AVOIDANCE RESULTS - 500M FIELD SIZE

No.	Waypoints	Conflicts	Near Misses	Distance Ratio
4	84.0	20.0	3.33	1.09
8	143.0	123.3	29.0	1.10
16	311.7	528.7	136.7	1.11
32	666.7	2448	596.7	1.11

TABLE VI
NO COLLISION AVOIDANCE RESULTS - 1000M FIELD SIZE

No.	Waypoints	Conflicts	Near Misses	Distance Ratio
4	39.7	10.0	1.0	1.04
8	92.0	43.3	12.3	1.05
16	185.7	179.7	44.7	1.05
32	378.7	665.0	159.3	1.05

TABLE VII
DONG ET AL METHODS - 500M FIELD SIZE

No.	Waypoints	Conflicts	Near Misses	Distance Ratio
4	53.3	8.0	3.3	1.81
8	71.7	58.3	14.3	2.50
16	72.3	225.0	54.0	4.72
32	56.7	686.3	172.7	7.30

TABLE VIII
DONG ET AL METHODS - 1000M FIELD SIZE

No.	Waypoints	Conflicts	Near Misses	Distance Ratio
4	32.0	7.0	1.3	1.36
8	59.3	10.0	4.0	1.52
16	107.7	355.0	27.3	1.86
32	110.3	428.7	111.3	2.77

TABLE IX
FINAL RESULTS - 500M FIELD SIZE

No.	Waypoints	Conflicts	Near Misses	Distance Ratio
4	62.7	0.0	0.0	1.67
8	82.3	0.0	0.0	1.08
16	76.0	14.7	2.3	4.27
32	50.0	64.0	12.3	7.55

TABLE X
FINAL RESULTS - 1000M FIELD SIZE

No.	Waypoints	Conflicts	Near Misses	Distance Ratio
4	39.7	0.0	0.0	1.08
8	80.3	0.0	0.0	1.17
16	128.7	1.7	0.3	1.51
32	160.3	21.0	3.0	2.07

A. Efficiency and Effectiveness

Two efficiency metrics were used to compare algorithms: the number of waypoints reached, and the distance ratio, α . The distance ratio is the ratio of the distances the UAV travelled to reach the waypoints over the minimum, straight-line travel distance.

$$\alpha = \frac{d_{actual}}{d_{min}} \quad (3)$$

The distance ratio only takes into account the distance travelled up to the last waypoint reached, thus this distance is not counted in α when the simulation ends. This is also true in calculating d_{min} . This metric shows how much the plane has deviated from a near-optimal path. The number of waypoints reached is also a metric on efficiency and helps to balance the limitations of the distance ratio metric. The planes attempt to reach all waypoints in order within the specified time limit. A plane may have a near perfect distance ratio, say it reaches one waypoint on a straight line path, but then fails to reach another waypoint during the simulation. Although for this individual UAV the distance ratio is near perfect, the number of waypoints reached will be significantly lower. Using waypoints reached counters the distance travelled that

the distance ratio fails to take into account as the UAVs travel at fixed speed.

As seen in the base case no avoidance results, the base case distance ratio is not a perfect 1.0, as d_{min} is the straight distance between waypoints, not the minimum flight path which would allow for the maximum turning radius. As expected, the distance ratio in high-density flight situations using a collision avoidance system is significantly higher than in the related no avoidance case. Similarly, the number of waypoints reached in the avoidance simulation results are significantly lower than in the no avoidance simulation results.

Effectiveness is a measure of conflict-reduction and near-miss reduction. Again, a *near-miss* is defined as one UAV traveling within a radius of 12 meters of another UAV. A note on these metrics: for this purpose, the main goal was to achieve zero collisions in each flight situation. Depending on the situation, waypoints may be given higher priority. By changing several of the rules in the collision avoidance fuzzy system to output less extreme changes in heading, more waypoints are achieved, the number of collisions increase slightly.

TABLE XI
DONG ET AL METHODS - 500M FIELD SIZE

No. Planes	Waypoints Reached	Distance Ratio	Near-Miss Reduction	Conflict Reduction
4	53.3	1.81	0.0%	60.0%
8	71.7	2.50	28.5%	52.7%
16	72.3	4.72	60.5%	57.4%
32	172.7	7.30	71.1%	72.0%

TABLE XII
DONG ET AL METHODS - 1000M FIELD SIZE

No. Planes	Waypoints Reached	Distance Ratio	Near-Miss Reduction	Conflict Reduction
4	32.0	1.36	0.0*	30.0%
8	59.3	1.52	67.50%	76.92%
16	107.7	1.86	38.93%	0.0*
32	110.3	2.77	30.3%	35.5%

(* denotes areas where Dongt et al. performed worse than the base case.)

This approach improved upon the base case in most categories and field sizes, but can be optimized further. Changes mentioned previously, such as adding the bearing angle of the opposite plane as a fuzzy input, were made to the methods of Dong et al and the results are detailed below. These are the results from the fuzzy rules and fuzzy set dimensions described in graphs and tables previously.

VI. CONCLUSIONS

This paper presents a solution to high-density UAV collision avoidance using fuzzy logic. While we still experienced near-misses in the higher density cases, the overall reduction in near-misses as compared to the base case was significant. In every field situation,

TABLE XIII
FINAL RESULTS - 500M FIELD SIZE

No. Planes	Waypoints Reached	Distance Ratio	Near-Miss Reduction	Conflict Reduction
4	62.7	1.67	100.0%	100.0%
8	82.3	1.08	100.0%	100.0%
16	76.0	4.27	98.32%	97.59%
32	50.0	7.55	97.94%	97.39%

TABLE XIV
FINAL RESULTS - 1000M FIELD SIZE

No. Planes	Waypoints Reached	Distance Ratio	Near-Miss Reduction	Conflict Reduction
4	39.7	1.08	100.0%	100.0%
8	80.3	1.17	100.0%	100.0%
16	128.7	1.51	99.33%	99.05%
32	160.3	2.07	98.12%	99.55%

we were able to decrease the number of near-misses compared to the base case by a minimum of 97%, 100% in the lower density cases. By using a simple determination mechanism to see if a UAV is in a collision situation, we use a fuzzy inference system to determine the appropriate change in heading based on the bearing angles of each plane and the distance between both planes. Results were generated by using a ROS based simulator which sends position updates for each UAV once per second. Results were evaluated based on effectiveness in avoiding collisions as well as efficiency in path taken and waypoints reached. The future work section details the areas of improvement that future research can take. Fuzzy logic systems should be considered as a valid solution to UAV collision avoidance.

VII. FUTURE WORK

One major advantage of a fuzzy logic system used in collision avoidance is dealing with uncertainty of inputs and outputs. In simulation, inputs are known exactly, but this is not always the case in real-world applications. Variance in data due to weather or noise could affect the certainty of data. Ranges of fuzzy sets will be able to handle this uncertainty of input measurement values as well as other system disturbances. A real-world implementation is part of intended future work of this research.

Other future work includes a better approach for determining the most dangerous UAV, or determining the UAV with which a collision is most imminent. As stated above, only the closest UAV to the current UAV was considered. Several geometric approaches exist that would be better suited to determining the most dangerous UAV, instead of the closest UAV. Exploring the use of a different determination mechanism would likely increase both efficiency and effectiveness. Limitations of our current approach include increased risk of collisions with planes flying relatively parallel or heading directly towards one another. One such approach that eliminates these two limitations is presented by Ghose et al. by using their *Zero Effort Miss* metric [20].

Future work will also include further optimization of fuzzy sets and rules. Time limited the number of test cases, and the sets and rules presented here are manually tuned. Other optimization techniques for fuzzy logic engines will be explored and implemented in the future.

REFERENCES

- [1] J. Holt, "Comparison of aerial collision avoidance algorithms in a simulated environment," Master's thesis, Auburn University, May 2012.
- [2] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *European Control Conference*. Citeseer, 2001, pp. 2603–2608.
- [3] B. Mettler and Z. Kong, "Receding horizon trajectory optimization with a finite-state value function approximation," in *American Control Conference, 2008*. IEEE, 2008, pp. 3810–3816.
- [4] B. Gardiner, W. Ahmad, T. Cooper, H. M., J. Holt, and S. Biaz, "Collision avoidance techniques for unmanned aerial vehicles," Tech. Rep., 2011.
- [5] T. Crescenzi, A. Kaizer, T. Young, J. Holt, and S. Biaz, "Collision avoidance in uavs using dynamic sparse a*," Tech. Rep., 2011.
- [6] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots," *Robotics Research*, pp. 365–374, 2005.
- [7] T. Liao, "Uav collision avoidance using a* algorithm," Ph.D. dissertation, Auburn University, 2012.
- [8] M. Park, J. Jeon, and M. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, vol. 3. Ieee, 2001, pp. 1530–1535.
- [9] H. Saradi. (2007, April) Local path planning using virtual potential field. [Online]. Available: <http://www.cs.mcgill.ca/~hsafad/robotics/index.html>
- [10] Y. Hwang and N. Ahuja, "A potential field approach to path planning," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 1, pp. 23–32, 1992.
- [11] K. Sigurd and J. How, "Uav trajectory design using total field collision avoidance," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Citeseer, 2003.
- [12] S. Han and H. Bang, "Proportional navigation-based optimal collision avoidance for uavs," in *2nd International Conference on Autonomous Robots and Agents*, 2004.
- [13] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 28, no. 5, pp. 562–574, 1998.
- [14] T. Ross and U. of New Mexico, *Fuzzy logic with engineering applications*. Wiley Online Library, 1997.
- [15] A. Martinez, E. Tunstel, and M. Jamshidi, "Fuzzy logic based collision avoidance for a mobile robot," in *Industrial Fuzzy Control and Intelligent Systems, 1993., IFIS'93., Third International Conference on*. IEEE, 1993, pp. 66–69.
- [16] N. Tsourveloudis, K. Valavanis, and T. Hebert, "Autonomous vehicle navigation utilizing electrostatic potential fields and fuzzy logic," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 4, pp. 490–497, 2001.
- [17] T. Dong, X. Liao, R. Zhang, Z. Sun, and Y. Song, "Path tracking and obstacles avoidance of uavs-fuzzy logic approach," in *Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on*. IEEE, 2005, pp. 43–48.
- [18] J. Manathara and D. Ghose, "Reactive collision avoidance of multiple realistic uavs," *Aircraft Engineering and Aerospace Technology*, vol. 83, no. 6, pp. 388–396, 2011.
- [19] P. Angelov, C. Bocaniala, C. Xideas, C. Patchett, D. Ansell, M. Everett, and G. Leng, "A passive approach to autonomous collision detection and avoidance in uninhabited aerial systems," in *Tenth International Conference on Computer Modeling and Simulation, 2008. UKSIM 2008.*, 2008.
- [20] J. G. Manathara, "Collision avoidance and coalition formation of multiple unmanned aerial vehicles in high density traffic environments," Master's thesis, Indian Institute of Science, May 2011.