UAV Collision Avoidance Using Artificial Potential Fields
Technical Report #CSSE11 - 03

Jason Ruchti
Robert Senkbeil
James Carroll
Jared Dickinson
James Holt
Saad Biaz

6 July 2011

Abstract

Over recent years, research into Unmanned Aerial Vehicles (UAVs) has increased substantially. However, if autonomous flight of these vehicles is to become feasible in the near future, a collision avoidance scheme must be developed to ensure airspace safety. In this paper, we propose a novel artificial potential field approach to this problem, which, combined with a powerful priority-based system, guides UAVs safely to their destinations.

1 Introduction

Unmanned aerial vehicles (shortened to UAVs) are becoming increasingly popular as the technology behind these machines improves and their capabilities grow. However, one of the chief concerns regarding UAVs is the possibility of collision. With no human controlling these vehicles, artificial intelligence becomes responsible for ensuring that the vehicle is not placed in a situation where a collision would occur. If the use of autonomous UAVs is to become feasible in the near future, a collision avoidance scheme must be developed that is able to (a) detect a potential conflict between multiple vehicles in an airspace, (b) present a viable path for the UAV to take in order to avoid collision and maintain a minimum separation distance, and (c) be intelligible to human operators in the same airspace.

Unlike ground-based vehicles, UAVs must deal with constant movement and limited turning ability, which makes collision avoidance much more complicated. However, even with the challenge of handling dynamic obstacles, there are quite a few methods available to handle collision avoidance. Unfortunately, the majority of these methods suffer from one of two issues. With many Calculus-based algorithms, processing complexity limits their usefulness as the time and computing capabilities needed become increasingly taxing from third-order or higher Bachmann-Landau performance rates [12]. With other geometry-based approaches, direction is more of a challenge as the algorithms can present directional changes that are too extreme for UAVs to handle.

Therefore, within this paper, a vector-based approach using artificial electric potential fields is detailed, specifically focusing on how collision avoidance can be implemented using charge attraction and repulsion alongside a powerful priority system. In addition, issues with sharp directional changes that are impossible for UAVs are addressed by introducing a circular-based waypoint range inspired by C-C-C and target touring minimum length paths [12]. This combination results in a powerful avoidance framework that successfully guides UAVs to their destinations: all accomplished with a time complexity of $O(n^2 log(n))$, where n is the number of vehicles in the airspace.

2 Problem Description

The overall goal of this research is to develop an algorithm to handle the collision avoidance of autonomous UAVs in a limited airspace. In an effort to further confine the problem, a few assumptions will be established. The airspace will be restricted to a two-dimensional plane, keeping the UAVs at a constant altitude. This restriction is incorporated to focus on developing an algorithm that performs soundly in two dimensions prior to opening the problem to three dimensions. In addition, the assumption will be made that each UAV will remain at a constant speed throughout the duration of its flight. This constraint was introduced in order to focus on designing an algorithm that successfully handles planes that are unable to speed up or slow down to avoid a collision, before the problem is expanded to allow for this ability. Finally, it will be assumed that each UAV is equipped with an autopilot that can successfully guide the UAVs to various destinations given waypoints specifying latitude, longitude, and altitude.

In order to evaluate the effectiveness of a collision avoidance algorithm, the qualifications that indicate varying levels of close encounters must be established. More specifically, two zones will be defined: an inner zone indicating a potential collision and a larger, outer zone indicating a conflict. Since the algorithm is intended to be used with modified Multiplex Easystar UAVs, the two zones have been defined based on the properties of these UAVs. The modified Easystar UAVs fly at an average speed of twenty-five miles per hour, or about eleven meters per second. As a result, the collision and conflict zones have been set at twelve and twenty-four meters respectively, approximating a one and two second buffer. The algorithm should minimize both the number of collisions and the amount of maneuvering needed to avoid other UAVs in the area in a way that maximizes the number of waypoints reached. Furthermore, because the hope is to eventually use the algorithm with UAVs in the field, it should take into account confounding factors such as network latency, packet loss, and the unique path characteristics of a UAV.

For testing purposes, the collision avoidance algorithm will be evaluated using a simulator. The simulator will allow three important advantages: it will eliminate unforseen variables associated with flying real UAVs such as the weather, it will allow stress testing of the algorithm to determine how many UAVs it can handle in a defined airspace without compromising physical UAVs, and it will provide a basis for which the algorithm can be compared against other collision avoidance algorithms in order to determine effectiveness. Using the simulator, eight different circumstances will be tested; plane counts of four, eight, sixteen, and thirty-two will be flown in airspaces of 500 by 500 meters and 1,000 by 1,000 meters. Each of these eight circumstances will be tested with three randomly generated courses lasting ten minutes.

3 Review of Various Approaches

3.1 Geometric Approach

One general method toward collision avoidance focuses on geometric algorithms such as the ones developed separately by Han, Bang, and Yoo [3] and Park, Oh, and Tahk [6]. For geometric algorithms, let UAVs be represented as point masses moving through airspace. Let a circle with a predefined radius be labeled as the obstacle zone such that for each UAV in the algorithm there is an obstacle zone that surrounds the UAV and indicates the area in which other UAVs must not enter. The algorithm works by determining the relative velocity vectors of the UAVs. If the relative velocity of one of the UAVs encroaches on another UAV's obstacle zone, as shown in Figure 1, corrective action is taken. The algorithm will attempt to create a new path that uses the smallest amount of time for collision avoidance. To achieve this goal, a cost function is defined to score the possible paths that could be taken to avoid collision. By minimizing this function, an optimal avoidance path is found, shown with the new vector, $\vec{e_r}$ in Figure 1.

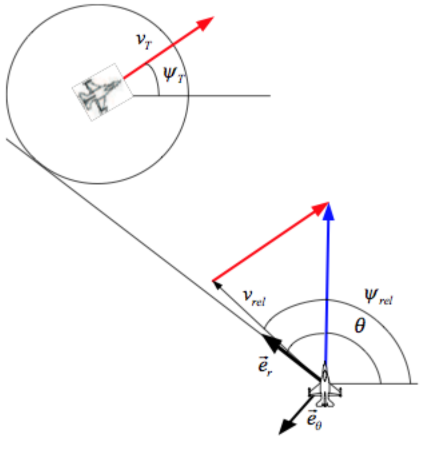

Figure 1: Collision Avoidance Using Geometric Approach [3]

The geometry-based approach works best in scenarios containing only two aerial vehicles as it is relatively straightforward and computationally efficient. However, in the case of multiple UAVs, calculations become more complicated. If there is overlap between the obstacle zones of the UAVs, the two UAVs are represented by a single point centered in the region where the obstacle zones overlap. Unfortunately, as the number of UAVs in the airspace increases, the mathematics involved to avoid collisions quickly becomes overwhelming, and performance deteriorates. It becomes difficult to program the intuitive geometric logic necessary to achieve a best path with multiple UAVs. Moreover, when the UAVs engage in collision avoidance, additional caution must be taken to ensure that the avoidance path is viable.

3.2 Evolutionary Algorithm Approach

Another interesting approach, developed by Rathbun, Kragelund, Pongpunwattana, and Capozzi [8], involves the use of evolutionary algorithms to avoid collisions between UAVs. The approach seeds a population of roughly twenty potential paths for a UAV; these paths are mixed to emulate reproduction or "mutated" through flight plan alterations. After the new generation is produced, their flight paths are scored by a cost function that takes into account the vehicle's flight capabilities, constraints on the paths, the flight environment, and the destination of the UAV. The paths with the best scores undergo a second iteration of mutation and reproduction while the other paths are discarded. This process continues for approximately fifty generations from which the best path is chosen.

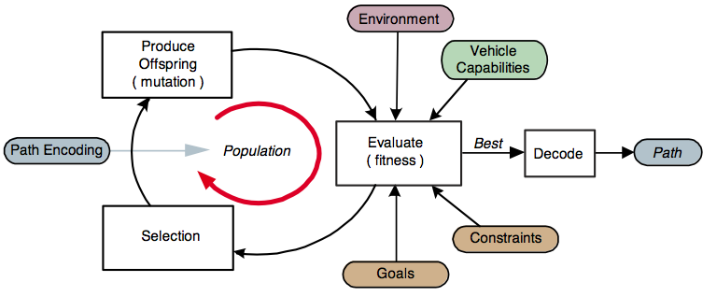

Figure 2: Overview of Evolutionary Algorithm [8]

Although the approach is based on random chance, the final flight plan chosen is often very efficient, requiring little excess maneuvering by the UAV. However, the cost of the randomness involved is extremely high due to the computational time and difficultly in the implementation. The cost function contains numerous restraints on the UAV's final path and therefore is extremely difficult to define and implement. Additionally, although the final path taken improves with each generation, the number of computations involved increases at an even higher rate. Because of this, there is a tradeoff between computation time and path viability as longer computation times yield better paths, but the amount of computation involved may prove unfeasible in a dynamic, real-time environment.

3.3 Grid-based Approach

Another approach, implemented by Alejo, Conde, Cobano, and Ollero [1], Meng and Gao [5], and Xia, Jun, Manyi, Ming, and Zhike [13], involves discretizing the flight space into a grid in order to perform graph algorithms such as A* and Dijkstra's algorithm. The first step in this approach involves the construction of the grid itself; however, because the Earth itself is curved, a small margin of error is introduced in the grid measurements. Each region in the grid is often represented as a node in a graph with arcs connecting it to adjacent grid spaces. Nodes can then be marked as forbidden or open to explore based on the position of other UAVs, obstacles in the flight path, and the flight characteristics of the UAV. The nodes in the grid are always at 45 degree angles from each other. Therefore, the flight path of the UAV must be represented in a different way, since planes do not always fly at angles that are multiples of 45. One solution is to the find the closest angle to the current bearing of the UAV that is a multiple of 45; the difference from the bearing to the closest angle is then given as a weight to the node in that grid space. The remainder is placed in the next closest square. In this way, the probability that the UAV will be in a certain grid space can be determined.

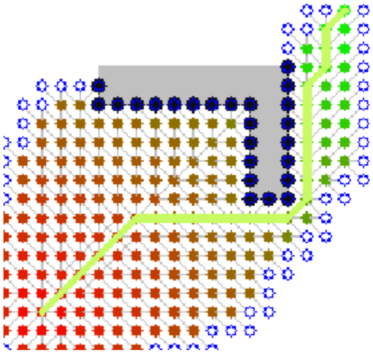

Figure 3: Classic A* Algorithm [2]

One variation of the standard A* algorithm, utilized in UAV path planning, is the Sparse A* algorithm. With Sparse A*, a smaller portion of the grid space is searched in order to improve efficiency. By specifying a minimum route leg length – a minimum distance that the UAV will be flying in a straight direction – and a maximum turning angle for the UAV, extraneous nodes can be pruned from the search [5]. Additionally, by limiting the depth of the search, efficiency can be improved at the cost of a slightly sub-optimal path. Moreover, because A* is a best-first search that computes a cost-function for various locations in the environment, the development of a good heuristic to incorporate as part of the cost function can greatly reduce the search time and improve effectiveness. With a good heuristic, computational complexity decreases from an exponential to polynomial

Bachmann-Landau efficiency.

Through grid formation and use of the Sparse A* algorithm, fairly optimal paths can be determined. When discretizing the grid space, a finer grid can yield better paths since the actual location of the UAV can be given with greater accuracy; however, the price for the this greater precision is computational time. Thus, a good balance between grid fineness and computation time must be found. Additionally, because A* was designed with static environments in mind, some problems may arise in a dynamic, real-time environment.

3.4 Mixed Integer Linear Programming Approach

Mixed Integer Linear Programming, also known as MILP, is another popular technique to solve the problem of UAV collision avoidance, and has proven to be effective by research teams lead by Jonathon How at MIT [9] [10]. Using MILP, a best outcome is found for a given mathematical model containing a set of linear constraints. To begin, a number of constraints need to be put in place to model the UAV's motion dynamics; additionally, constraints related to collision avoidance such as minimum separation distance must be defined. These constraints are then adapted for commercial MILP solving tools such as APML and MATLAB, which develop a best path for the UAV in the airspace based on the provided constraints. Each aircraft's path is highly optimized based on the expected paths of the other UAVs in the airspace; therefore, an uncooperative UAV can wreak havoc on the carefully planned system.

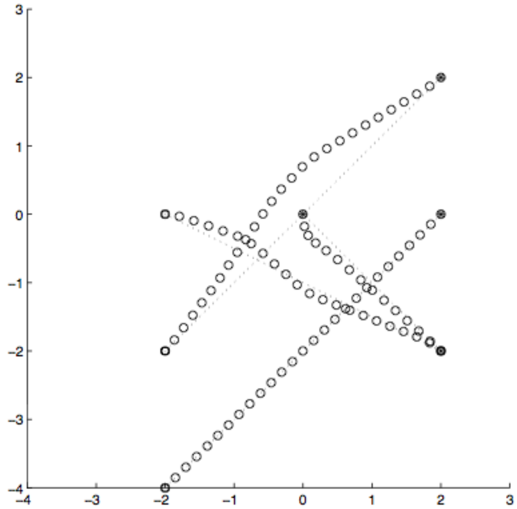

Figure 4: A path for four UAVs using MILP. The white circles denote the path traveled to the destination, represented by a star inscribed within a circle. [9]

The solution to the mathematical model specified often involves trade-offs between optimality and computation time. Because MILP solving belongs to the NP-hard class of problems, this approach rapidly becomes intractable when multiple UAVs become involved, especially when a fixed arrival time approach is used. Using the fixed arrival time approach solves the problem over a given time range and is often extremely computationally expensive; however, a technique known as receding horizon helps reduce this problem. Using a receding horizon approach, the UAV path is broken down into small chunks where the MILP model is solved for N time steps, which serve as input for the next N time steps and so on until the process is finished. While this approach minimizes the computational burden, since the program is solved in smaller blocks, obstacles outside one time block distance from the UAV will have no effect on the path of the UAV; therefore, when paths for the next time block need to be computed, the UAVs may be too close to avoid collision. Thus, when using the MILP approach, the goal is to minimize computation time while increasing the size of the time blocks used in computation.

4 Artificial Electric Potential Fields

4.1 Relevant Research

Our research focuses on the use of artificial electric potential fields to ensure UAV collision avoidance. The inspiration for this approach comes from research including Liu, Wang, and Dissanayake's use of artificial force fields in robot collision avoidance [4], Andrew Proctor's work on UAV collision avoidance using artificial potential fields [7], and Karin Sigurd and Jonathon How's approach to UAV collision avoidance using total fields [11].

Liu, Wang, and Dissanayake's research on artificial force fields in robot collision avoidance solved the path finding problem for an omnidirectional class of vehicles. By modeling goal waypoints as positive charges and other robots in the project space as negative charges, optimal routes were found by calculating a total force from the attractive force of the robots towards their destination and repulsive force away from other vehicles operating in the area. One particularly important innovation is the use of the elliptical potential fields, which resulted in less effort for the robot to continue forward since the artificial field generated by a robot extends further into the space ahead of it. Another important innovation included the use of the expanding force fields; as a robot moved closer to its goal waypoint, the artificial field increased in size to ward off other robots in the area. Additionally, special cases including the deadlock problem – two robots are approaching each other head-on with destinations directly behind each other – are discussed. Their proposed soution was to have one robot turn to the left or right to avoid a head-on collision.

Proctor's research demonstrated the efficacy of the artificial potential field method for scenarios in which there were small numbers of UAVs in the airspace, and he also presented a solution to the deadlock problem. Like Liu, Wang, and Dissanayake, an attractive force was found between the UAV and its destination; additionally, the summation of all the repulsive forces acting on the UAV was calculated. The total force was calculated by the addition of the attractive force and the total repulsive force. However, unlike robots, UAVs cannot turn sharply or come to a complete stop – an issue that Proctor's research did not take into account. While Proctor's research resulted in a working solution to simple UAV collision avoidance, further research needed to be done on scenarios with multiple UAVs and multiple waypoints.

In Sigurd and How's research, physical magnetic fields were used to prevent UAV collisions. To test this idea, powerful magnets along with magnetic field sensors were placed on each of the UAVs prior to testing. The magnetic sensors were configured to cancel the effects of Earth's magnetic field in addition to the magnetic field generated by the powerful magnet attached to its respective UAV. The goal waypoints of the UAVs were modeled with a strong north magnetism, while each of the UAVs exhibited a strong south magnetism. In this way, the UAVs were attracted to the destination while repelled from each other. As with Proctor's research, a total force determined from attractive and repulsive forces was calculated for each UAV. One additional innovation was the attraction constant, γ, $0 < \gamma < 1$. After the total attractive and repulsive forces acting on a UAV were determined, their magnitudes were scaled to unit vectors. The final force acting on the UAV was calculated using the following equation:

$$\vec{F_{tot}} = \gamma * \vec{F_{attr}} + (1 - \gamma) * \vec{F_{rep}}$$

The use of the attraction constant, γ, ensured that minimum weight was given to the attractive force; therefore, it could be safely assumed that the UAV would be progressing towards its destination. Their results indicated that an attractive constant of approximately 0.66 yielded aggressive, yet safe, flight paths.

4.2 Motivation

The artificial potential field approach offers a number of distinct advantages when compared to other methods. First, unlike MILP, grid-based, and evolutionary algorithms, using artificial potential fields is not computationally expensive; therefore, it can be used with older computers, microprocessors, and other devices with limited proccessing capabilities. This reduction in computational complexity makes the artificial potential field approach tractable for large numbers of UAVs. Moreover, because the force on each UAV is calculated independently, the problem could potentially be distributed onto microprocessors aboard each of the UAVs, which further reduces the computational burden on any single machine. Additionally, since the UAVs operate independently, uncooperative aircraft in the airspace can be handled without issues.

4.3 Our Approach

Our technique for UAV collision avoidance melds the approaches of the researchers previously discussed with certain novel features aimed to solve unique challenges with UAV path planning. We model the UAVs as negatively charged particles of magnitude q_{UAV}, whereas the destinations hold a positive charge of magnitude q_{dest}. For several reasons, a naive approach of emulating the electrical force from physics leads to suboptimal results. A powerful repulsive force often dominates over the attractive force in scenarios containing multiple UAVs causing the planes to never reach their destinations. Additionally, when the artificial field extends to infinity, UAVs that should be safe from collision may be affected by unnecessary repulsive forces. Thus, the challenge is to modify the electrical force in ways that are most conducive to UAV flight. Based on our research, the modifications should include the following changes: limiting the resulting force vector to lie at an angle no greater than the maximum turning angle of the plane, defining a boundary for each plane beyond which the effects of the force field are not felt, and altering the force calculations so they act linearly with distance instead of quadratically.

4.3.1 Determining the Force Acting on a UAV

In order to begin calculations, we must first define the characteristics of the artificial potential field. Although electric fields in nature extend infinitely, for our purposes a finite field prevents UAVs from being affected by unnecessary repulsive forces, yielding more optimal results. Therefore, we first define a constant d_{onesec} representing the distance a UAV can travel in one second, and a circle of radius d_{onesec} around the current position of the UAV that defines the collision zone no other UAV should enter. From this baseline, we define α as the ratio of the collision zone to the size of the artificial potential field generated by the UAV; experimentally, we have found best results with $\alpha = 5$. We have determined that artificial potential fields generated by the UAVs should be elliptical rather than circular in shape, so that the potential fields extend further in front of each UAV to repel other vehicles in immediate danger of collision. Let λ_{scalef} be a scalar that extends the field in front of the UAV, and λ_{scaleb} be a scalar that lessens the field behind a UAV. The following formula defines the maximum distance of the field:

$$d_{fmax} = \alpha * d_{onesec}[(\lambda_{scalef} - (\lambda_{scalef} - \lambda_{scaleb})/2)] + ((\lambda_{scalef} - \lambda_{scaleb})/2) * cos(\theta))$$

where θ is the angle from the bearing of another UAV, U_k, to the location of the current UAV, U_i. We have found best results experimentally with $\lambda_{scalef} = 2$ and $\lambda_{scaleb} = 1.25$.

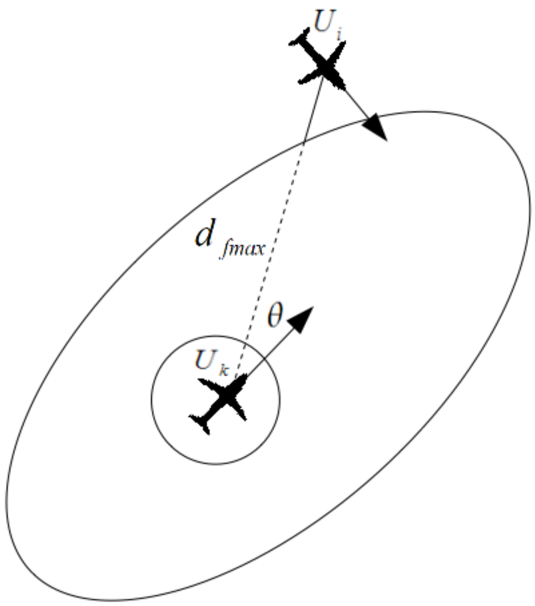

Figure 5: Determining the maximum distance of the repulsive force field

To further protect the forward airspace of the UAV, the repulsive force is modified to generate a greater force directly in front of the UAV. In our approach, a polar coordinate system is used to describe the force vectors. The vectors with the greatest magnitude will have the greatest impact on the final bearing of the UAV. To calculate the repulsive force acting on a UAV, we find the angle θ from the bearing of a UAV, U_k, to the location of another UAV, U_i. We define the variable d_{fmax} as the maximum distance the potential field acts over the given angle, θ; it is calculated using the equation previously defined. Moreover, we define d as the current distance between U_i and U_k. If $d > d_{fmax}$, no repulsive force is present since the distance between the UAVs is greater than the field generated; otherwise, a repulsive force must be calculated.

Let $d_{failsafe}$ define a critical distance between two planes that should not be breached, and let $F_{failsafe}$ define an arbitrarily large force value. If $d \leq d_{failsafe}$, we assign a force of $F_{failsafe}$ to U_i as a failsafe to avoid collision if the planes get too close to each other. The magnitude of $F_{failsafe}$ is so large that it overpowers any other forces present, and therefore, the two airplanes can both focus solely on moving out of each other's way. We have defined $d_{failsafe}$ as the distance traveled by our UAVs in 2.5 seconds, and $F_{failsafe}$ as 9999.

Now, if $d_{failsafe} < d \leq d_{fmax}$, we must determine a repulsive force emitted by U_k to act on U_i. The magnitude of the force is based first and foremost on the distance between the two UAVs. Additionally, we modify the repulsive force to emit a stronger force in front of U_k than behind it. To begin, we define two constants. Let k_{emitf} be a scalar to increase the force emitted in front of U_k, and k_{emitb} be a scalar to decrease the force emitted in back of the UAV. Additionally, let γ be a positive scalar to serve as a check on the strength of the repulsive force. To find the repulsive force emitted by a UAV, U_k, to act on another UAV,

U_i, the following formula is employed:

$$r(\theta) = \begin{cases} 0 & \text{if } d > d_{fmax} \\ F_{failsafe} & \text{if } d \leq d_{failsafe} \\ q_{UAV} * [(k_{emitf} - (k_{emitf} - k_{emitb})/2)] + ((k_{emitf} - k_{emitb})/2) * cos(\theta)^{\frac{d_{fmax}-d}{\gamma * \alpha}} & \text{else} \end{cases}$$

Experimentally, we have found best results with $q_{UAV} = 80$, $k_{emitf} = 1.5$, $k_{emitb} = 1$, and $\gamma = 4$.

Additionally, since the planes are moving at a constant speed, if the repulsive force felt by a UAV is coming from behind, it should play a smaller role in the final force calculation, since the UAV is already moving away. Therefore, we first find the angle ϕ_{rep} between the current bearing of U_i and the repulsive force acting on U_i from U_k. Now, let β_{feelf} be a scaler to increase the force felt from a UAV emitting from the front of a UAV, U_i, and let β_{feelb} be a scaler to decrease the force felt from a UAV emitting from behind. Additionally, let θ be the angle from the bearing of a UAV, U_k, to the location of another UAV, U_i, as defined above. To find the force felt by U_i from U_k, the following formula is used:

$$s(\theta, \phi_{rep}) = r(\theta) * [(\beta_{feelf} - (\beta_{feelf} - \beta_{feelr})/2) - ((\beta_{feelf} - \beta_{feelr})/2) * cos(\phi_{rep})]$$

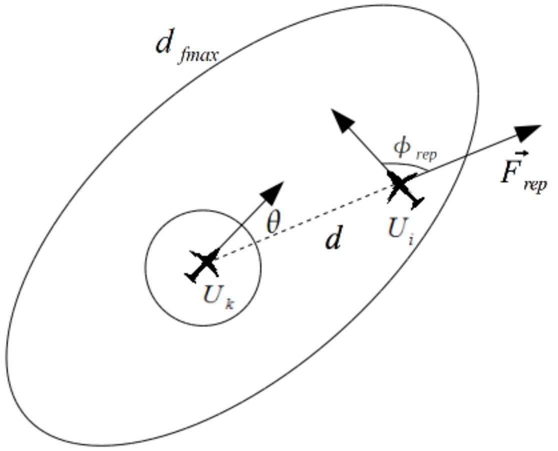

Figure 6: Geometry involved with calculating the repulsive force acting on U_i

Experimentally, we have found best results with $\beta_{feelf} = 1$ and $\beta_{feelb} = 0.5$.

By using this function, the repulsive force felt is lessened for UAVs approaching from behind and strengthened for UAVs approaching from the front. To find the total repulsive force acting on the UAV, U_i, we must find the sum of the repulsive forces acting on U_i by the other n UAVs in the airspace using the formula:

$$\vec{F_{rep}} = \sum_{k=1}^{n} s(\theta_k, \phi_{repk})$$

where θ_k is the angle between the location of U_i and the bearing of U_k, and ϕ_{repk} is the angle between the bearing of U_i and the repulsive force acting on U_i from U_k.

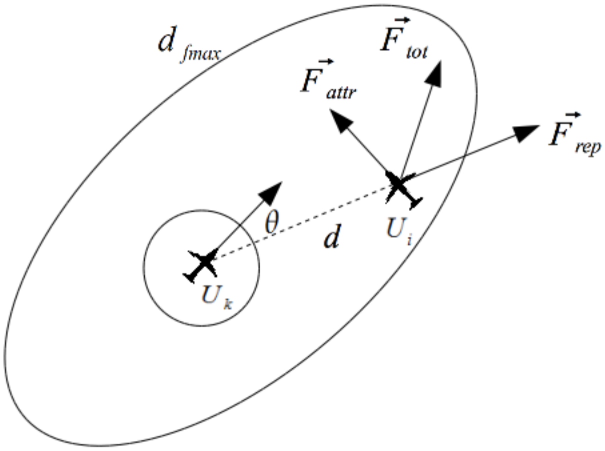

Figure 7: Calculating the total force acting on a UAV

In our approach, the attractive force, $\vec{F_{attr}}$, is a constant; best results are found with $|\vec{F_{attr}}| = 100$. The total force acting on the UAV can be determined using the equation $\vec{F_{tot}} = \vec{F_{attr}} + \vec{F_{rep}}$. However, simply turning in the direction of the new force is not viable; therefore, let $\Delta\theta$ be the angle between $\vec{F_{tot}}$ and the current bearing of the UAV, θ_i. A maximum turning angle, $\Delta\theta_{max}$, is specified if $\Delta\theta > \Delta\theta_{max}$, then $\Delta\theta$ is scaled down to $\Delta\theta_{max}$. Additionally, if $\Delta\theta < -\Delta\theta_{max}$, then $\Delta\theta$ is scaled up to $-\Delta\theta_{max}$. For our tests, $\Delta\theta_{max}$ was set at 22.5 degrees. In this way, the next destination of the UAV is limited to a circular-based range of points one second away that deviate no more than $\Delta\theta_{max}$ from the current bearing of the UAV.

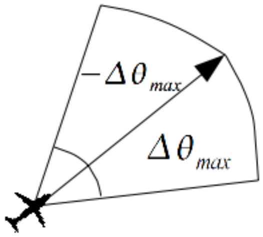

Figure 8: Circular-based range of points reachable by the UAV in one second

4.3.2 Deadlock Case

A special case occurs when two UAVs are approaching head-on towards each other with waypoints directly behind each other. In this case, the repulsion force is pointed directly behind each UAV; no motion to the left or right is present to break the deadlock. To determine if a head-on collision is about to occur, we first find unit vectors for the attractive and repulsive forces. If $\overrightarrow{F_{attrunit}} + \overrightarrow{F_{repunit}} = 0$, a head-on collision is about to occur. Using the naive approach of strictly emulating electric forces, the UAVs would continue straight ahead until a crash most likely occurred. To avoid this catastrophe, the UAVs are given a directional change of 15 degrees right in order to break the deadlock and allow normal force calculations to resume, as shown in Figure 9.

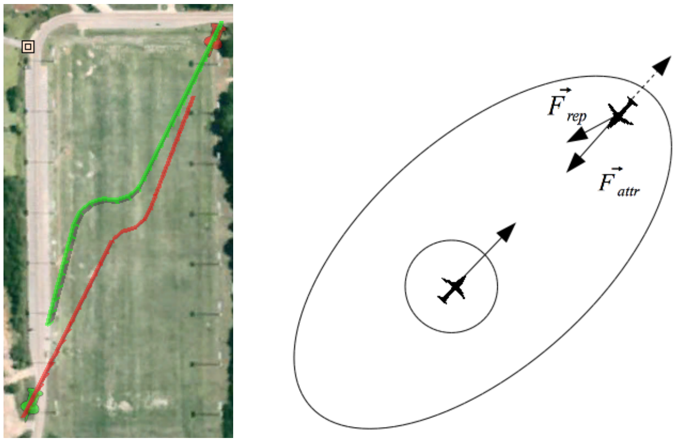

Figure 9: Adjusting the repulsive force to handle head-on collisions

4.3.3 Right Hand Turn Rule

In UAV flight, it is sometimes more advantageous for a UAV to cross behind another UAV in the airspace than in front of it; therefore, some geometric techniques are employed to force this type of maneuver. To begin, we find the angle, θ, between the bearing of another UAV, U_k, and the location of the current UAV, U_i. If θ is on the interval $[-135, 0]$, U_i is to the left of U_k, and we proceed to the next step. Otherwise, no corrective action needs to be taken. For the next step, we find the angle ϕ_{rep} between the current bearing of U_i and the repulsive force acting on U_i from U_k. If ϕ_{rep} is on the interval $[-180, -90]$, the UAV is attempting to make a left hand turn to avoid collision; however, such a maneuver will place it in front of U_k. Therefore, the angle of the repulsive force may need adjustment.

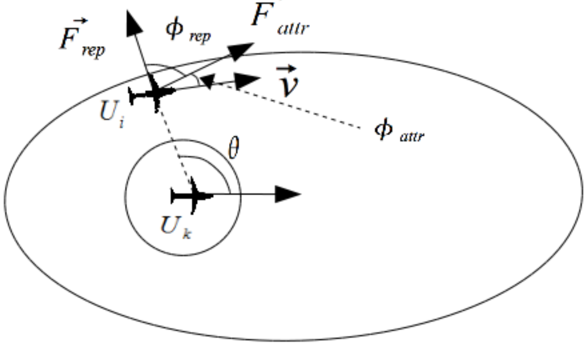

Figure 10: Situation in which the right hand turn rule should not be applied. U_i should continue towards its destination instead of traveling behind U_k.

To begin, let ϕ_{attr}, be the angle between the bearing of U_i and the attractive force it feels from its destination. If $\phi_{attr} \leq 0$ and $\theta < -90$, we do not force a right hand turn, since this plane should make a left turn to head towards its destination, as shown in Figure 11. Otherwise, if θ is on the interval $[-135, -25]$, a right hand turn should be forced so the UAVs cross behind each other. Therefore, the initial angle ϕ_{rep} is rotated across the length of the UAV to arrive at ϕ_{repf}, as shown in Figure 11. For other angles on the interval $[-25, 0]$, we first find the spherical triangle relating to current geometry of the UAVs. To begin, let a be the great-circle distance between the two UAVs, U_i and U_k. Let b be the great-circle distance between U_i and its point of intersection with U_k, and let c be the great-circle distance between U_k and its point of intersection with U_i. Each side of this spherical triangle has an associated angle, A, B, and C, as shown in Figure 11. Now, if $(c - b) \leq (-\phi_{repi} - 90)$, we force a right hand turn by rotating the initial angle ϕ_{rep} across the length of the UAV to arrive at ϕ_{repf}, as shown in Figure 11.

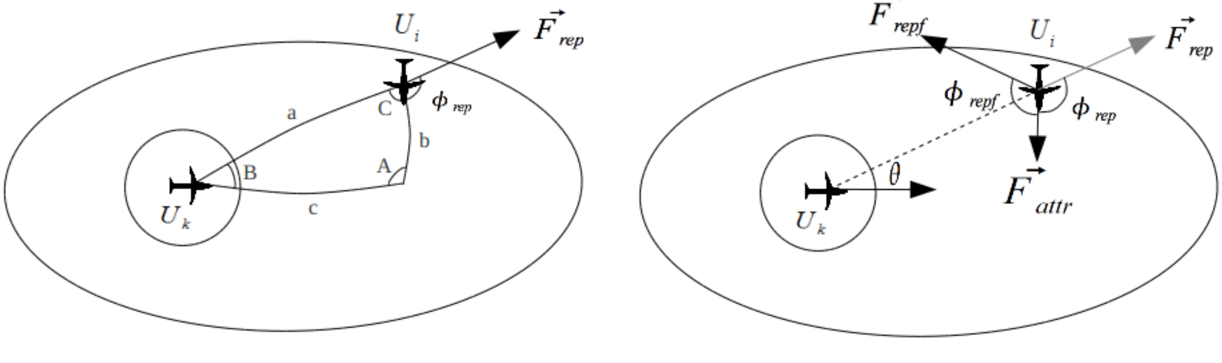

Figure 11: Finding the spherical triangle and the new repulsive force vector in order to travel behind U_k

4.3.4 Priorities

A quirk in our design is related to the maximum turning angle of the UAV. As a UAV, U_i, approaches its destination, it feels an attractive force inducing it to continue forward; however, other UAVs present may generate a repulsive force that pushes U_i away from its destination. This causes unnecessary motion as U_i is forced to make excess turns to come back to its destination after almost reaching it. However, a far more serious problem is also apparent. As the other UAVs continue on toward their destinations, the repulsive force will gradually dissipate to zero; however, U_i may be displaced from the goal in such a way that it cannot make the turn necessary to reach the current waypoint and move onto the next destination. Because of this, the UAV becomes stuck in a centripetal motion about its destination.

To solve this problem, we designed a system of priorities to steer UAVs directly toward their destinations while still preventing collisions. As a UAV approaches its destination, it is given a priority once it is within a distance of $d_{priority}$ from its destination; we let $d_{priority} = 4.5 * d_{onesec}$. Priorities are used to allow UAVs closer to their destinations to move less affected by repulsive forces; UAVs with higher priorities ignore forces from UAVs with lower priorities. Consider a system of UAVs in a limited airspace. To calculate the force acting on a UAV, U_i, within a distance of $d_{priority}$ from its destination and with priority m, we must find the vector sum of the repulsive forces it feels from those UAVs with a higher priority; mathematically this can be expressed using the equation

$$\vec{F_{rep}} = \sum_{k=1}^{m-1} s(\theta_k, \phi_{repk})$$

where θ_k is the angle between the bearing of another UAV, U_k, and the location of U_i and ϕ_{repk} is the angle between the bearing of U_i and the repulsive force it feels from U_k.

The addition of priorities removes a great deal of excess maneuvering when a UAV is close to its destination; however, the priority system raises a new issue. Normally, when two planes without priority encounter each other in the airspace, both make an effort to get out of each other's way. With the priority system, only one UAV makes an effort to avoid collision. Thus, the field generated by a plane with priority should expand farther to give other planes in the airspace more time to make the turns necessary to avoid collision. To achieve this, we define a scaler, $p_{mult} > 1$, as a multiplier to expand distance of the force field for those planes with priority. Thus, if a plane has priority, its expanded field can be determined by the equation $p_{mult} * d_{fmax}$, as shown in Figure 12 (recall that d_{fmax} is the maximum distance the force field of a UAV acts over). We have found best results with $p_{mult} = 1.2$.

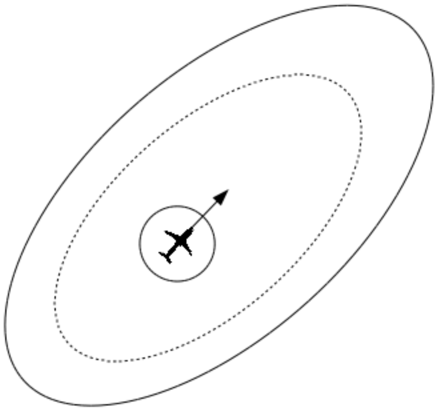

Figure 12: Example of expanding fields. The dotted line represents the old maximum field distance before expansion.

4.3.5 Detection and Correction of Looping

However, a priority system does not fully remove the issue of looping. If a situation arises where two waypoints are within close proximity to one another, the UAV will become stuck in centripetal motion after it reaches the first waypoint and attempts to make the turn necessary to achieve the second waypoint. To solve this problem, an algorithm has been developed to recognize the looping condition and break the centripetal motion. To begin, we must find the radius of the circle created when the UAV engages in a 360 degree turn. In order to achieve this, we use the formula $s = r_{turn}\theta$; solving for r_{turn}, we find $r_{turn} = \frac{s}{\theta}$. Now, let θ be the maximum turning angle of the UAV, so that $s \approx d_{onesec}$. For the UAVs used in our tests, $r_{turn} = \frac{11.176m}{22.5*\frac{\pi}{180}}$, yielding $r_{turn} \approx 28.46m$, the distance traveled by the UAV in approximately 2.5 seconds.

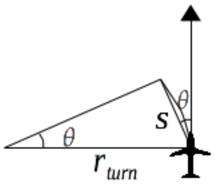

Figure 13: Determining the turning radius of the UAV

In our current system, a UAV is designated as having reached its waypoint if it is within a threshold distance, d_{thres}, from its destination. Therefore, as shown in Figure 14, there is a circular region of points that a UAV cannot reach if it is undergoing a loop. The maximum distance between a looping UAV and its waypoint is $d_{loopmax}$, as shown in Figure 14, $d_{loopmax} = 2 * r_{turn} - d_{thres}$. When the UAV is within $d_{loopmax}$ meters from its waypoint, we begin to monitor for looping. To begin, we must first find the location of the center of the circle generated when the UAV engages in a centripetal motion around its waypoint at a maximum turn angle. Generally, for actual planes in constant motion, this center is located a distance r_{turn} away from the current position, 90 degrees to the UAV's left or right, depending on whether the UAV is engaging in a left or right turn, respectively. However, because the simulator only updates once per second, our looping path is a hexadecagon as opposed to a circle. This causes the center of the loop to lie at an angle of 101.25 degrees from the current bearing of the plane. Next, we calculate the radius of the circle delineating the unreachable waypoint zone, as shown in Figure 14, $r_{unreachable} = r_{turn} - d_{thres}$. Next, we find the distance, $d_{ctodest}$, between the UAV's destination and the center of the circle generated when the UAV makes a loop around its destination.

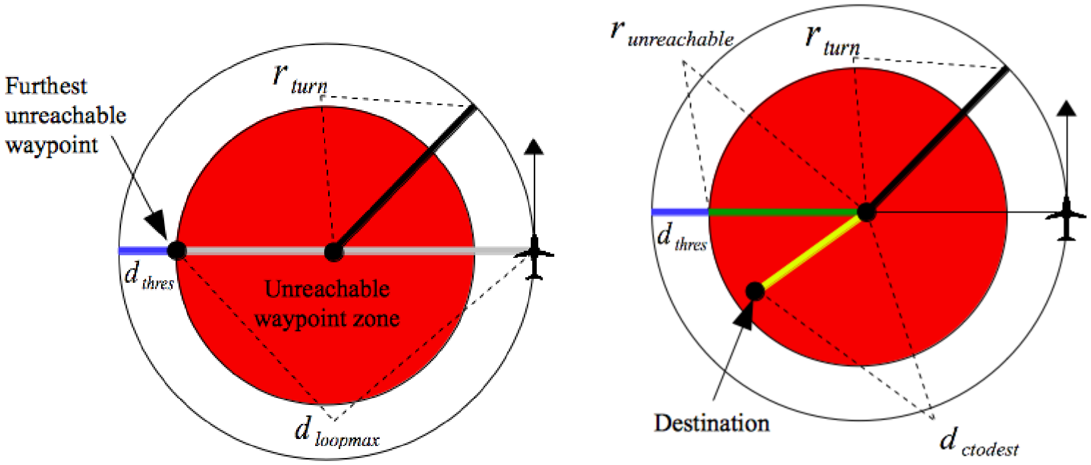

Figure 14: Geometry involved to detect a looping condition.

If $d_{ctodest} > r_{unreachable}$, the destination lies outside of the unreachable waypoint zone, and the UAV can continue without correction. Otherwise, to correct this condition, the destination is modeled as a repulsive force until the UAV's distance to destination, d_{dest}, is greater than $d_{loopmax}$ in order to push the UAV out of the loop, as shown in Figure 15.

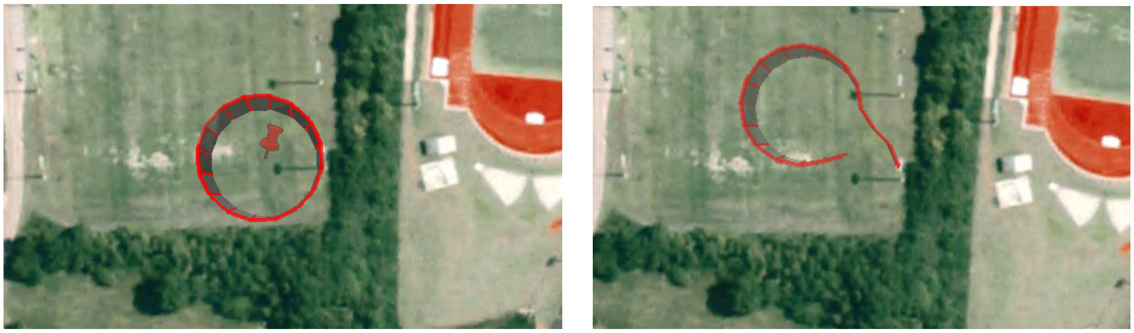

Figure 15: Demonstration of the looping detection and correction mechanism.

4.4 Results

To test our implementation of collision avoidance using artificial potential fields, we used a simulator to model the airplanes' flight characteristics. As with any simulation, there are some limitations; however, we believe the model accurately portrays conditions in the real world. To begin, random course files were generated giving each plane fifty waypoints to achieve over ten minutes. Three course files were generated for scenarios with four, eight, sixteen, and thirty-two planes within a stressful 500 by 500 meter airspace and a more realistic 1000 by 1000 meter airspace. We defined a conflict zone as the distance traveled by the UAV over two seconds, and a potential collision zone as the distance traveled by the UAV over one second. If two UAVs entered into a potential collision, they were deleted from the scenario to model the real world. However, it should be noted that a potential collision does not imply an actual collision as the size of the UAVs are minimal compared to the size of the potential collision zone.

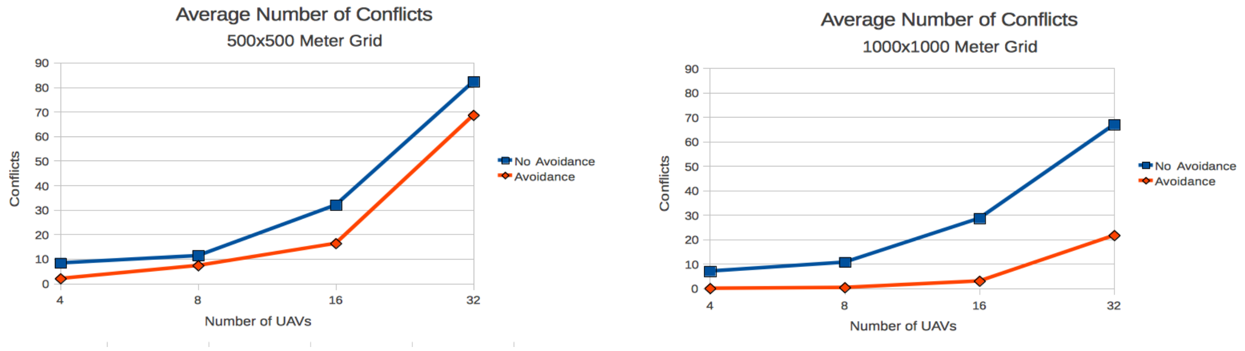

Figure 16: Comparision of the number of conflicts.

Tests without collision avoidance were performed as a baseline for comparison against our algorithm. As shown in Figures 16 and 17, our algorithm performs well compared to the baseline. Inside a stressful 500 by 500 meter field, our algorithm was effective for scenarios containing four and eight planes. Additionally, in the more realistic 1000 by 1000 meter field, the scenarios with four, eight, and sixteen planes fair extremely well. However, as shown, some situations prove infeasible. Within a 500 by 500 meter field, our algorithm quickly deteriorates with 32 UAVs. However, the algorithm quickly stabilizes after shedding a number of UAVs within a few seconds of starting and performs reasonably after about 100 seconds.

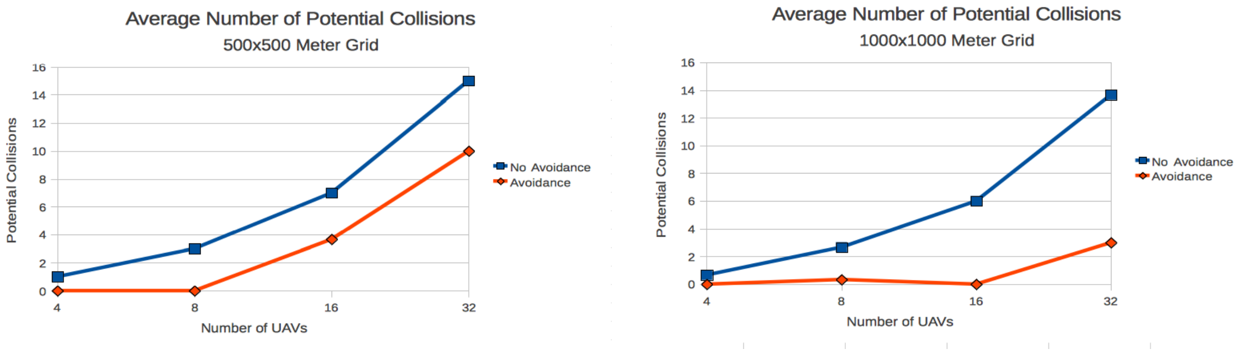

Figure 17: Comparision of the number of potential collisions.

Another metric, the ratio of the actual path distance to the straight-line path distance, tests the efficiency of our algorithm. The straight-line path measures the sum of the distance between each of the waypoints if perfectly straight paths were taken; as such, the ratio will be greater than one even without collision avoidance as the UAVs are restricted by a turning angle as they move toward their next destination. In the stressful 500 by 500 meter environment, the large ratio values indicate that a large number of moves were dedicated towards collision avoidance; however, even this was not enough to stop an inordinate amount of potential collisions in the scenario with 32 UAVs. Therefore, we believe many of these potential collisions were unavoidable. However, in the more realistic 1000 by 1000 meter field, the ratio is much more reasonable, indicating our algorithm succeeds in reducing the number of excess moves required to avoid collisions.

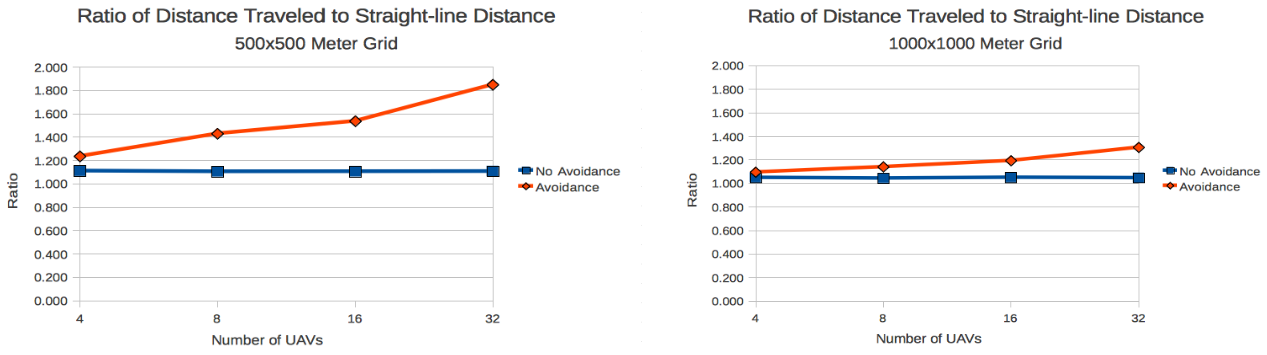

Figure 18: Comparision of the ratio of actual path distance to straight-line path.

Another important metric is the number of waypoints achieved over the course of the simulation. As shown, all of our scenarios demonstrated improvement over the baseline. Because collisions were averted, the UAVs were able to continue on their paths and achieve more waypoints.

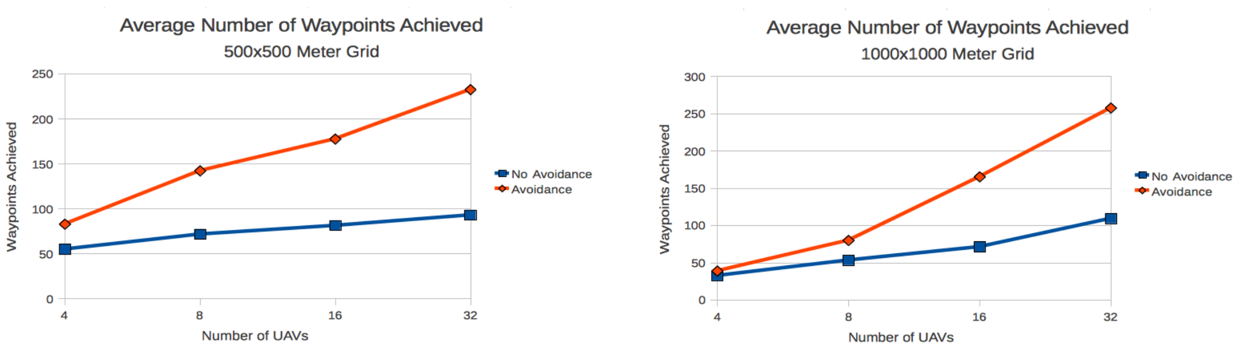

Figure 19: Comparision of the number of waypoints achieved.

From our experimental data, we believe our algorithm represents a strong choice for collision avoidance. By minimizing the ratio of total distance traveled to straight-line distance, our algorithm demonstrates its efficiency in reducing the amount of motion necessary to avoid collisions. In realistic scenarios, our algorithm demonstrates strength when compared to the baseline, often reducing the number of conflicts and potential collisions to near zero. From the data, we believe our algorithm is best suited for scenarios where the plane density is approximately eight UAVs per 500 square meters; in these situations, our algorithm has demonstrated significant strength. This plane density provides a large buffer over what would be expected in the real flight, where a density of two to three UAVs per 500 square meters would be much more common.

5 Conclusion

We have proposed a novel solution to collision avoidance using artificial electric potential fields to plan UAV motion. Through the use of artificial electric potential fields, we enforce a safe distance between the UAVs, while modeling destinations as attractive forces ensures that the UAVs will continue traveling toward their goals. Furthermore, we ensure path viability through the use of priority-based systems along with confining the turning angle to a circular-based range of points. The minimization of computational cost to a mere $O(n^2 log(n))$, where n is the number of UAVs in the airspace, indicates an improvement over other techniques discussed such as the evolutionary algorithm, MILP, or grid-based approaches. It allows for significant scalability to hundreds or thousands of autonomous UAVs given enough airspace. As a result, we believe that artificial electric potential fields present a very realistic solution for UAV collision avoidance.

References

[1] D. Alejo, R. Conde, J.A. Cobano, and A. Ollero. Multi-UAV Collision Avoidance with Separation Assurance Under Uncertainties. In *Mechatronics, 2009. ICM 2009. IEEE*

International Conference on, pages 1–6, April 2009.

[2] Subhrajit Bhattacharya. Classic A* Algorithm.

[3] Su-Cheol Han, Hyochoong Bang, and Chang-Sun Yoo. Proportional Navigation-based Collision Avoidance for UAVs. *International Journal of Control Automation and Systems*, 7:553–565, 2009.

[4] D.K. Liu, D. Wang, and G. Dissanayake. A Force Field Method based Multi-Robot Collaboration. In *Robotics, Automation and Mechatronics, 2006 IEEE Conference on*, pages 1–6, June 2006.

[5] Bobo Meng and Xiaoguang Gao. UAV Path Planning based on Bidirectional Sparse A* Search Algorithm. In *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, volume 3, pages 1106–1109, May 2010.

[6] Jung-Woo Park, Hyon-Dong Oh, and Min-Jea Tahk. UAV Collision Avoidance based on Geometric Approach. In *SICE Annual Conference, 2008*, pages 2122–2126, Aug. 2008.

[7] Andrew Proctor. Development of an Avoidance Algorithm for Multiple, Autonomous UAVs in a Finite Space, June 2010.

[8] D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi. An Evolution based Path Planning Algorithm for Autonomous Motion of a UAV through Uncertain Environments. In *Digital Avionics Systems Conference, 2002. Proceedings. The 21st*, volume 2, pages 8D2:1–12 vol.2, 2002.

[9] A. Richards and J.P. How. Aircraft Trajectory Planning with Collision Avoidance using Mixed Integer Linear Programming. In *American Control Conference, 2002. Proceedings of the 2002*, volume 3, pages 1936–1941, 2002.

[10] Tom Schouwenaars, Bart De Moor, Eric Feron, and Jonathan How. Mixed Integer Programming for Multi-vehicle Path Planning. In *American Control Conference, 2001. Proceedings of the 2001*, 2001.

[11] Karin Sigurd and Jonathan How. UAV Trajectory Design using Total Field Collision Avoidance. *American Institute of Aeronautics and Astronautics*, 2003.

[12] Hong Wong, V. Kapila, and R. Vaidyanathan. UAV Optimal Path Planning using C-C-C Class Paths for Target Touring. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 1, pages 1105–1110 Vol.1, Dec. 2004.

[13] Li Xia, Xie Jun, Cai Manyi, Xie Ming, and Wang Zhike. Path Planning for UAV based on Improved Heuristic A*; Algorithm. In *Electronic Measurement Instruments, 2009. ICEMI '09. 9th International Conference on*, pages 3:488–493, Aug. 2009.