2010

Improving particle swarm optimization path planning through inclusion of flight mechanics

Joseph Scott Holub
Iowa State University

Improving particle swarm optimization path planning through inclusion of flight mechanics

by

Joseph Scott Holub

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-majors: Mechanical Engineering; Human Computer Interaction

Program of Study Committee:
Eliot Winer, Co-major Professor
Song Zhang
Stephen Gilbert

Iowa State University

Ames, Iowa

2010

Table of Contents

List of Figures iii

List of Tables v

Abstract vi

Chapter 1 – Introduction 1

Chapter 2 –Background 21

Chapter 3 – Method Development 41

Chapter 4 – Results and Evaluation 69

Chapter 5 – Conclusions and Future Work 93

Acknowledgements 98

Bibliography 99

List of Figures

FIGURE 1: PREDATOR UAV 1

FIGURE 2: X-45 UAV (BOEING PHANTOM) 1

FIGURE 3: PREDATOR GROUND CONTROL STATION 3

FIGURE 4: UAV SINGLE CAMERA VIEW 4

FIGURE 5: MUSCIT CONTROL STATION 5

FIGURE 6: OPTIMIZATION DESIGN SPACE EXAMPLE 9

FIGURE 7: 2D UAV FLIGHT PATH CONTROL INTERFACE 15

FIGURE 8: VIRTUAL BATTLESPACE PROJECT VISUALIZATION OF UAV FLIGHT PATHS 16

FIGURE 9: MICROSOFT'S FLIGHT SIMULATOR X 17

FIGURE 10: THE IMPACT OF INCLUDING FLIGHT DYNAMICS IN PATH PLANNING 19

FIGURE 11: VIRTUAL BATTLESPACE PROJECT 21

FIGURE 12: LOGITECH CORDLESS RUMBLEPAD 2 22

FIGURE 13: VIRTUAL BATTLESPACE PATH PLANNER VISUALIZATION OF THE UAV PATH 23

FIGURE 14: PITCH, ROLL, AND YAW OF AN AIRCRAFT 29

FIGURE 15: BALANCED FORCES ON AN AIRCRAFT 30

FIGURE 16: B-SPLINE REPRESENTATION WITH CONTROL POINTS (P) 33

FIGURE 17: INTERPOLATED B-SPLINE REPRESENTATION 34

FIGURE 18: 2D PATH PLANNER DISPLAYING ALTITUDE 36

FIGURE 19: 3D PATH PLANNER SHOWING ISOMETRIC VIEW 36

FIGURE 20: 2D PATH PLANNER SHOWING ALTERNATE PATHS 37

FIGURE 21: VIRTUAL BATTLESPACE META-PATHS VISUALIZATION 38

FIGURE 22: VIRTUAL BATTLESPACE'S PATH PLANNING FLOWCHART 41

FIGURE 23: THREAT AVOIDANCE REPRESENTED IN 2D 43

FIGURE 24: RECONNAISSANCE DIAGRAM 49

FIGURE 25: PULL-OUT CONDITION FREE BODY DIAGRAM 57

FIGURE 26: CLIMBING CONDITION FREE BODY DIAGRAM 58

FIGURE 27: GLIDING CONDITION FREE BODY DIAGRAM 59

FIGURE 28: STEADY, LEVEL, CO-ORDINATE TURN FREE BODY DIAGRAM 61

FIGURE 29: ORIGINAL VISUALIZATION OF ALTERNATE PATHS 66

FIGURE 30: 2D REPRESENTATION OF DIFFERENCE BETWEEN ORIGINAL VISUALIZATION AND B-SPLINE INTERPOLATION 67

FIGURE 31: SCENARIO 2, WITH ONE GROUND THREAT 71

FIGURE 32: SCENARIO 3, WITH ONE AERIAL THREAT 71

FIGURE 33: SCENARIO 4, WITH ONE GROUND AND ONE AERIAL THREAT 72

FIGURE 34: SCENARIO 5, WITH TWO GROUND THREATS AND ONE AERIAL THREAT 72

FIGURE 35: SCENARIO 2 SIDE VIEW 77

FIGURE 36: SCENARIO 2 TOP VIEW 77

FIGURE 37: ORIGINAL PSO PATH PLANNER RESULTS 78

FIGURE 38: PSO PATH PLANNER WITH FLIGHT MECHANICS 78

FIGURE 39: ORIGINAL PATH PLANNER VISUALIZATION OF ALTERNATE PATHS 90

FIGURE 40: NEW PATH PLANNER VISUALIZATION OF ALTERNATE PATH USING B-SPLINE CURVES 91

List of Tables

TABLE 1: LIST OF EVALUATION SCENARIOS 70

TABLE 2: UAV AIRCRAFT SPECIFICATIONS FOR EVALUATION 74

TABLE 3: COMPONENT WEIGHTS 74

TABLE 4: BREAKDOWN OF COMPONENT COSTS FOR SCENARIO 2 RUN 1 WITHOUT FLIGHT MECHANICS 80

TABLE 5: BREAKDOWN OF COMPONENT COSTS FOR SCENARIO 2 RUN 5 WITH FLIGHT MECHANICS 81

TABLE 6: AVERAGE TOTAL COST VALUES FOR SCENARIO 2 83

TABLE 7: AVERAGE TOTALS FOR EIGHT RUNS OF SCENARIO 2 FOR THREE DIFFERENT TYPES OF PATHS 84

TABLE 8: AVERAGE TOTAL COST OF ALL FIVE SCENARIOS CONSIDERING THREE TYPES OF PATHS, 8 RUNS EACH 86

TABLE 9: AVERAGE TOTAL COST FOR THREE TYPES OF PATHS COMPARING WITH FLIGHT MECHANICS AND WITHOUT FLIGHT MECHANICS 88

Abstract

Military engagements are continuing the movement toward automated and unmanned vehicles for a variety of simple and complex tasks. This allows humans to stay away from dangerous situations and use their skills for more difficult tasks. One important piece of this strategy is the use of automated path planners for unmanned aerial vehicles (UAVs). Current UAV operation requires multiple individuals to control a single plane, tying up important human resources. Often paths are planned by creating waypoints for a vehicle to fly through, with the intention of doing reconnaissance while avoiding as much danger to the plane as possible. Path planners often plan routes without taking into consideration the UAV's ability to perform the maneuvers required to fly the specified waypoints, instead relying upon them to fly as close as possible.

This thesis presents a path planner solution incorporating vehicle mechanics to insure feasible flight paths. This path planner uses Particle Swarm Optimization (PSO) and digital pheromones to generate multiple three-dimensional flight paths for the operator to choose from. B-spline curves are generated using universal interpolation with each path waypoint representing a control point. The b-spline curve represents the flight path of the UAV. Each point along the curve is evaluated for fuel efficiency, threat avoidance, reconnaissance, terrain avoidance, and vehicle mechanics.

Optimization of the flight path occurs based on operator defined performance

characteristics, such as maximum threat avoidance or minimum vehicle dynamics cost. These performance characteristics can be defined for each unique aircraft, allowing the same formulation to be used for any aircraft. The vehicle mechanics conditions considered are pull-out, glide, climb, and steady, level, co-ordinate turns. Calculating the flight mechanics requires knowing the velocity and angle of the plane, calculated using the derivative of the point on the curve. The flight mechanics of the path allows the path planner to determine whether the path exceeds the maximum load factor (G-force), minimum velocity (stall velocity), or the minimum turning radius. Comparing the results between PSO Path Planner with flight mechanics and PSO Path Planner without flight mechanics over five scenarios indicates an increase in the feasibility of the returned paths.

Visualizing the flight paths was improved by changing the original waypoint based visualization to a b-spline curve representation. Using b-spline curves allows for an accurate representation of the actual UAV flight path especially when considering turns. Operators no longer must create a mental representation of the flight path to match the waypoints.

Chapter 1 – Introduction

Military Usage of Unmanned Systems

Military operations have begun using various types of unmanned systems for various tasks considered too dangerous for human operators, such as surveillance or disposal of improvised explosive devices (IEDs). Unmanned systems have taken many forms in different mediums, from ground vehicles, surface water vehicles, submersibles, and aerial vehicles. The push toward unmanned systems has involved many layers of government including the United States Congress and the President of the United States [1]. Unmanned systems have the potential to reduce the total defense cost as well as human loss, but only if they can reliably perform all types of combat mission either in lieu of or alongside human operators. The potential for these systems are widely recognized by many countries around the world.

One type of unmanned system, Unmanned Aerial Vehicles (UAVs), has shown potential for reliably performing all types of combat missions. For example, the Predator (Figure 1)[2] and the X-45 (Figure 2)[3] are currently two of the most commonly used UAV systems by the US Military. According to the Associated

Figure 1: Predator UAV

Figure 2: X-45 UAV (Boeing Phantom)

Press, as of January 2nd, 2008, UAVs have flown more than 500,000 mission hours, mostly in Iraq [4]. Between January and October, the military doubled their UAV usage, and consequently repositioning active pilots from the cockpit to the UAV base stations. Having pilots control UAVs instead of flying missions from the cockpit demonstrates the government's conscious recognition of the importance of UAVs in military operations. The Department of Defense has found that the use of UAVs in military operations has many possible positive implications and therefore created a 25-year roadmap to guide the development of unmanned systems [5]. The roadmap outlines four specific areas for unmanned systems research: Reconnaissance and Surveillance, Target Identification and Designation, Counter-Mine warfare, and Chemical, Biological, Radiological, Nuclear, Explosive (CBRNE) Reconnaissance. The plan envisions a force of UAVs that are both more autonomous than current systems and can seamlessly integrate with both manned and unmanned forces. Moving from a predominately manned force to an unmanned force would shift pilots from controlling planes in the cockpits to supervising multiple UAVs from a military ground control station. This paradigm shift results in the challenge of creating a control station interface that is intuitive and complementary to the pilot's knowledge and experience.

UAV Command and Control

Currently, all UAVs used in military operations are at least partially controlled by ground control stations (GCS). Figure 3 illustrates a Predator GCS, which is one type

Figure 3: Predator Ground Control Station

of unmanned vehicle control station used for large vehicles [6]. The Predator requires two operators for a single UAV, a trend seen across many types of UAVs. Controlling the Predator is extremely taxing on both operators. Therefore, operators are forced to switch every few hours with another pair of operators. Furthermore, each operator is trained for specific tasks. One operator must have full flight training to pilot the Predator. The other operator is trained to use the Predator's sensors.

Ground control stations are designed to mimic cockpits the pilot's are more familiar with. The pilot's interface shows the plane's statistics (speed, heading, altitude, etc.), a 2D map of the region (current position, known enemies' positions, friendly's positions). This information is standard for all planes and UAVs with the exception of a single camera video feed from the nose of the UAV that provides the remote pilot a soda straw type view of their environment. The soda straw view limits the pilot's situational awareness of the environment around them, Figure 4 [7]. Situational

awareness, the perception of environmental elements within a space, is key to a single operator controlling multiple UAVs effectively in military operations.

Figure 4: UAV single camera view

The ground control station interface must be considered given the US Military's strong focus on expanding the unmanned force. The current interface requires constant control by two trained operators, with one being a flight-trained pilot. Increasing the UAV force would require at least two more pilots to control each UAV, thus increasing the need for UAV pilots. The lack of pilots is one problem with the current paradigm. To achieve the Department of Defense's goal of many UAVs controlled by a single person, the whole control and interface of UAVs must be changed.

The Air Force Research Laboratory has attempted to solve this problem by developing the Multi-UAV Supervisory Control Interface Technology (MUSCIT) shown in Figure 5 [8]. MUSCIT allows pilots to control multiple Predators by displaying four camera views as well as a 2D situational view of the battlefield on a single computer. Pilots must still face the challenge of matching individual UAV camera views with the correct UAV in the 2D situational view to understand what is happening. The soda straw view and the 2D situational view inhibit the pilot's ability

Figure 5: MUSCIT control station

to maintain situational awareness. The MUSCIT system and other similar systems highlight the importance of autonomy in controlling multiple UAVs.

UAVs can make routine decisions without requiring constant operator input with higher levels of autonomy. The MUSCIT system is designed to control four UAVs simultaneously. However, if all four UAVs require the operator's attention at the same moment, multitasking and task switching will tax the operator's working memory and cognitive resources. Researchers have investigated working memory during cognitive tests and found a person's performance decreases as the number of simultaneous tasks increases [9][10][11][12]. UAVs must have a sufficient level of autonomy to allow the operator to focus on addressing priority tasks until lower priority UAV alert's can be addressed. The more autonomous a vehicle, the less time it needs to be monitored and therefore an operator can control multiple vehicles. In

order for the automation to be effective the operator must have some level of trust in its performance. Without trust, the operator will spend too much time monitoring specifics of the flight path, such as heading, altitude, and speed, when no intervention is required. However, if there is a level of trust developed between the operator and the automation system, the operator will be less likely to micromanage and intervene only when necessary [13]. In the case of path planning, the operator must trust the algorithm to return feasible paths or they will not use the system. Ideally, the operator would be provided with enough information and situational awareness that they would foresee when intervention is necessary with a specific UAV. While this would be ideal, the challenge is in designing a system that requires intervention at only the appropriate times.

Path Planning

There are many ways to make UAVs more autonomous including automated takeoff, automated landing, target recognition, target following, and path planning. Path planning is the process of calculating a travel route based on acquired information, such as a threat or target to investigate [14]. Autonomous path planning is vital to achieving the Department of Defense's goal of higher UAV utilization because computer analysis of the data with a visual display of the options will require less human capital than an individual analyzing and designing new paths. In today's battlefield of the technological age, new information about targets and threats are always streaming in during an engagement. Creating UAVs that can generate

acceptable paths to avoid threats and investigate new targets would free up the operator to control other aspects of the engagement and prevent unnecessary loss of UAVs.

Path planning is a complex problem of finding acceptable travel paths by evaluating possible paths against predefined criteria, such as safety, fuel efficiency, or reconnaissance. Path planning is divided into two types, online or offline. Online path planning is the act of planning an alternate path while the UAV is flying the mission a threat is encountered. Online path planning is difficult to achieve because it must operate close to real time to be effective. Offline path planning is completed before the UAV leaves the ground and is often calculated on a different computer than the one controlling the UAV. Offline planners are often more accurate but take more time to calculate a desired path. While finding a path to avoid threats could be considered simple, such as flying great distances to avoid potential threats, an optimal path balancing reconnaissance, fuel efficiency, and threat avoidance is more desirable. Determining a single optimal path from thousands of possibilities can be computationally expensive even for simple path constraints. As computers become faster, online path planners can calculate more complex path constraints.

Different path planners attempt to solve various issues associated with path planning. There is currently no complete solution to the problems associated with UAV integration. One problem being addressed is swarm management of multiple UAVs in a small area. Swarm management is important in order to avoid collisions

when multiple UAVs are present in the same area [15][16]. An additional issue is planning optimal paths, for example finding the shortest path between multiple waypoints. Some considerations in optimal path planning include using less fuel, avoiding threats, or minimizing radar cross section [17]. Researchers are currently exploring multiple mathematical equations, optimization methods, and optimization problem formulations to solve this complex system of equations, such as Genetic Algorithms and Particle Swarm Optimization [14][18][19][20][21][22]. Dynamic path planning has proven beneficial when new or uncertain information is provided [23][24][25]. The utilization of vehicle mechanics and dynamics are being explored to ensure the paths are feasible for the UAV [26][27][28].

This research will focus on incorporating flight mechanics into an online particle swarm optimization path planner. The issue of planning feasible paths based on vehicle characteristics such as velocity, flying altitude, turning radius, and load factor (force on the plane) will be addressed. Path planners that utilize flight dynamics and mechanics often settle for a velocity range and minimum turning radius. While this captures some constraints of a UAV, it does not take into consideration the forces acting on the UAV, the effective stall velocity as a UAV turns, or the change in minimum turning radius based on the speed of the UAV. Without these constraints, path planners can yield unfeasible flight paths.

Optimization

Optimization is referred to as the process of finding the best solution to a problem however the problem is defined [29]. In the case of path planning, optimization would involve finding the best feasible flight path to achieve maximum reconnaissance while avoiding all threats. Optimization problems seek to either minimize or maximize a given function within the parameters of the problem. One example of an optimization design space is illustrated in Figure 6 with the individual design variables on the x and y axes and the objective function value (F(x,y)) on the z-axis [30].

The blue arc represents the objective function values across the design space. In this optimization problem, we want to find the maximum value of the objective function for this design space. The optimum value is located at the top of the dome, or the highest point on the z-axis. In the case of path planning, the design space would be the collection of all possible routes the UAV could travel, with the maximum point being the best possible path to take given a set of criteria.

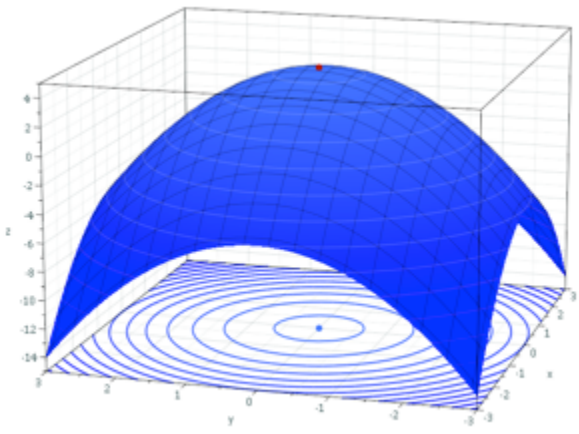

Figure 6: Optimization design space example

There are many different areas of optimization using unique techniques such as Newton's

method, hill climbing, genetic algorithms, or particle swarm optimization. Path planning is a multi-modal optimization problem, which means there can be multiple paths taken to achieve the same goals. Multi-modal optimization problems like path planning are often solved using heuristic methods, such as genetic algorithms (GA) [31], simulated annealing (SA) [32], or particle swarm optimization (PSO) [33]. Heuristic methods use a population of potential solutions to explore the design space in an attempt to find the optimal solution [34].

While all heuristic methods for optimization will return high-quality solutions to unconstrained multi-modal problems, like path planning, PSO was the most computationally efficient [35]. Particle swarm optimization is a popular technique that mimics the behavior or birds flocking or fish swimming. This technique randomly generates a population of particles within the design space. Consider bees collecting pollen, each particle would represent a bee and the design space would be a field of flowers. The bees are randomly placed in the field. The algorithm then moves the bees randomly within the field. The bees want to gather pollen, so are attracted to areas of the field with more pollen. Each particle (bee) can randomly pass good areas (pollen rich) and they can hear of the best area found by the entire swarm. Using their knowledge of the best area they have seen, the knowledge of best area found by the entire swarm, and their current traveling direction, the bee modifies its direction in the hopes of finding the best location. In PSO path planning, each

particle of the swarm represents a single possible path that could be taken by the UAV. Therefore the swarm is trying to find the optimal path for the UAV to travel.

The basic PSO algorithm is outlined in Equations 1, 2, and 3. The first equation is the velocity vector update equation, which controls the direction the particle will move in the design space. The direction and velocity the particle will travel is based on its current velocity vector, previous best location (pBest, best position found by that particle), and the global best location (gBest, best position of all particles). Equation 2 shows how the inertial weight factor is controlled. The inertial weight factor controls the level of contribution from the particles current velocity vector to the new velocity vector. Equation 3 is the position update equation, which combines the particle's position with the new velocity to determine the particle's new location in the design space.

$$V_{iter+1} = w_{iter}V_{iter,i} + c_1 rand_p(\ )[pBest_i(\ ) - X_i(\ )] + c_2 rand_g(\ )[gBest_i(\ ) - X_i(\ )] \quad (1)$$

$$w_{iter+1} = w_{iter}\lambda_w \quad (2)$$

$$X_{iter+1} = X_{iter} + V_{iter+1} \quad (3)$$

Swarms can also use pheromones, chemical scents, to direct the swarm in positive directions. Consider ants swarming food. Initially a single ant (particle) explores the area looking for food. When the ant finds food, it leaves a pheromone trail back to the nest. More ants will start to follow the pheromone trail as they come upon it. The

more ants follow the trail, the more ants drop pheromones on their way back to the nest making the pheromone trail stronger. When a better area is found the pheromone trail to this location dissipates due to the decrease in ants dropping pheromones. This process, defined as digital pheromones, can be used in Particle Swarm Optimization to improve swarming characteristics.

Digital pheromones mimic natural pheromones in most respects, such as increasing in strength the more pheromones dropped in an area or dissipating after no pheromones have been dropped in a while. The original PSO formulation works well, but the pBest and gBest information is inefficient. Along with the original velocity vector formulation, digital pheromones add another velocity component directed toward the closest and strongest pheromone, providing a more efficient method [36].

Equation 4 gives us the pheromone attraction factor (P') calculated using the distance between the particle and the pheromone (d) and the strength of the pheromone (P). The pheromone attraction factor determines which pheromone ($TargetPheromone_i$), the particle will target when calculating a new velocity vector (V_{iter+i}) as shown in Equation 5. The distance between the pheromone (X_{pk}) and the particle (X_k) is calculated using Equation 6, where the distance between upper and lower limits of the kth design variable is given as (range) and there are (k) number of design variables.

$$P' = (1 - d)P \tag{4}$$

$$\begin{aligned} V_{iter+1} = {} & w_{iter}V_{iter,i} + c_1 rand_p(\ \ )[pBest_i(\ \ ) - X_i(\ \ )] \\ & + c_2 rand_g(\ \ )[gBest_i(\ \ ) - X_i(\ \ )] \\ & + c_3 rand_T(\ \ )[TargetPheromone_i(\ \ ) - X_i(\ \ )] \end{aligned} \tag{5}$$

$$d = \sqrt{\sum_{l}^{k} \left(\frac{X_{p_k} - X_k}{range_k}\right)^2} \tag{6}$$

This research uses a particle swarm optimization algorithm developed at Iowa State University to balance three constraints, fuel efficiency, reconnaissance, and threat avoidance [37]. This thesis will discuss the modifications made to this algorithm through adding another constraint for flight mechanics. Each constraint acts as a cost function to push the path away from infeasible areas and toward optimal areas.

NURBS/B-splines

In this application we generate a smooth NURBS curve to define the path taken by a UAV. The waypoints the UAV flies through are treated as control points for the curve. By moving a single control point the flight path (the curve) can be changed.

Non-uniform rational basis spline (NURBS) is a way to mathematically representing curves and surfaces in computer modeling. NURBS are becoming one of the most used methods for representing curves and surfaces because of their many positive characteristics. Designing with NURBS is considered intuitive because almost every tool has a geometric interpretation. NURBS algorithms are fast and numerically

stable. The curves and surfaces do not vary when common geometric transformation, such as translation, rotation, and scale, are applied [38].

NURBS can be generated with a specific set of information, namely a list of control points, a knot vector, weights, and the order of the curve. Each point on the NURBS curve is found by computing the weighted sum of each control point. The control points determine the shape of the curve. The knot vector is a sequence of values that determine which control points affect each point on the curve. Changing the knot vector will change the shape of a curve while using the exact same control points. The order of the curve is the final information needed to define a curve. The order determines how many of the neighboring control points influence a point on the curve. The higher the order, the more control points influence the location of the curve point (e.g. an order of 3 will have 4 control points influencing each point on the curve). Using the knot vector and order of the curve, a set of basis functions can be generated. The point on the curve is found by multiplying the control points by their corresponding basis functions.

Visualization

Visualization is an important aspect in maintaining situational awareness, especially in the transition to one operator controlling multiple UAVs simultaneously. Currently, UAV operators can see 2D map views of the GPS location relative to the map. The only visual information received from the UAV is a soda straw view of what the UAV's camera sees. Figure 7 shows a typical interface with the blue icon indicating

the UAV and its direction [39]. The window in the upper right corner of Figure 7 shows the UAV flight dynamics information, such as the speed, location, and heading. Another option for visualization is to use a virtual world to show the entire physical scene. The Virtual Battlespace project does this by taking terrain information and information about units in the area and constructing a virtual scene [40]. Figure 8 illustrates an example of the virtual world constructed to view UAV command and control. By using a virtual representation, the operator can look at the scene from any position and retain situational awareness.

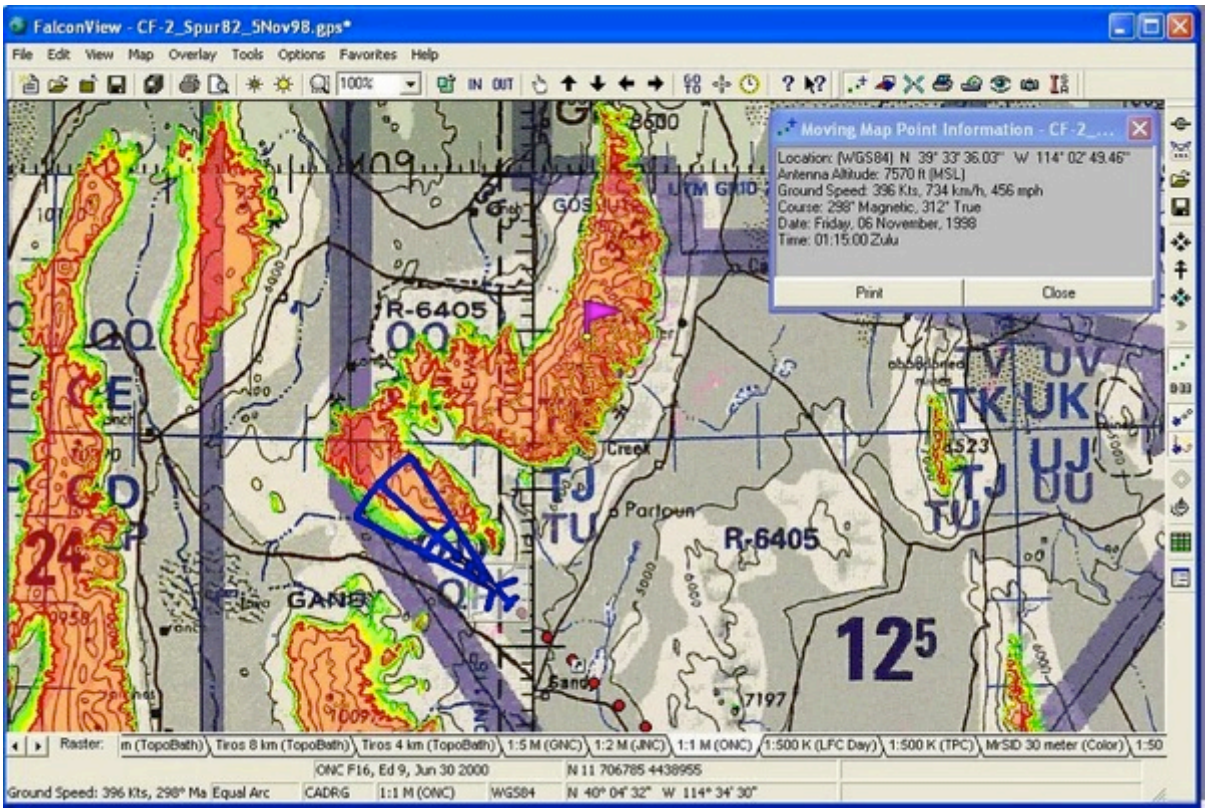

Figure 7: 2D UAV flight path control interface

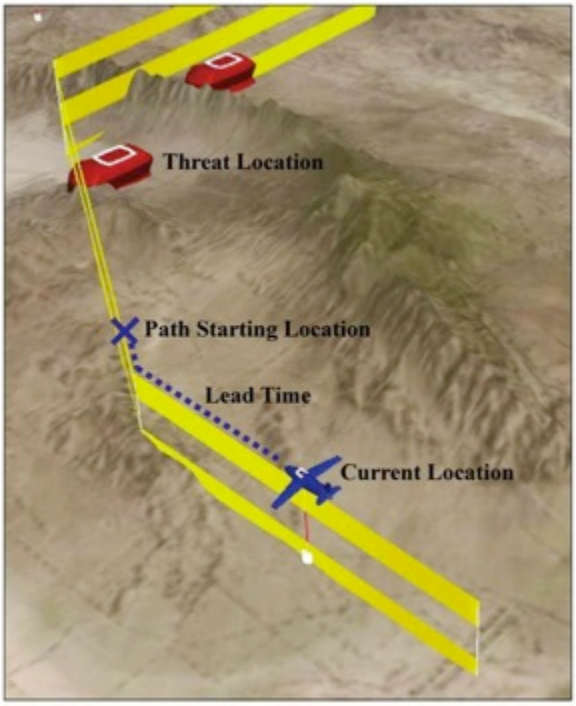

Figure 8: Virtual Battlespace project visualization of UAV flight paths

There are many different ideas for how to approach visualization of UAV command and control, the most commonly used is a version similar to Figure 6. The U.S. military currently relies on multiple pilots to control a single UAV with large control stations like the example in Figure 3. To expand the capabilities of a single operator to control multiple UAVs, consideration must be given to the interface and visualization of the information provided to the operator. Operators must maintain situational awareness at all times to effectively control multiple UAVs at the same time.

Vehicle and Fight Simulators

Simulators are well known in many industries including manufacturing, product design, and training. Using sophisticated computer models to replace physical objects, a simulator is able to replicate the physical object's capabilities and limitations. Simulators are currently being used for everything from manufacturing simulations, to racecar simulations, and flight training for pilots. Microsoft Corporation has been developing their "Flight Simulator" franchise for 25 years [41]. Flight Simulator X, shown in Figure 9, provides the user with a very realistic

Figure 9: Microsoft's Flight Simulator X

environment with accurate control models and responses. With the rapid increases in computational power, most driving/flying video games have basic simulators built into them, so the technology is becoming more accessible to everyone. Simulators are often used in training because they provide valuable feedback/experience to the user while providing a safe environment.

The United States military uses flight simulators to train pilots because they offer a safe training ground for the pilots and limit the training costs of using multi-million dollar vehicles. Flight simulations for the military include things as easy as taking off and landing, to more sophisticated procedures like how to fly in battle, how to recover in an emergency, or how to coordinate air support with ground troops. The

goal of all path planners is to provide simulator accuracy paths in real time. Computational power is the limit on achieving this goal. Evaluating flight models for thousands of possible paths requires too much time for the dynamic planning required of online path planners.

The goal of this project is to use basic vehicle flight mechanics to improve upon the path planning algorithm currently used. Flight mechanics will simulate basic aircraft flight characteristics and ensure the paths being planned are feasible for the given aircraft. Incorporating flight mechanics will move the path planner one step closer to the goal of simulator type accuracy.

Motivation

There has been much work done on creating autonomous path planning algorithms but there currently is no comprehensive solution. Path planning is broken up into two fundamental areas, the first being the algorithms used to generate feasible paths, and the second being visualizing the information to the operator in a clear manner. This thesis will cover both of these areas, with the focus being on the problem formulation of the path planner.

One aspect of the Battlespace path planner that has not been addressed is flight dynamics. While the current optimization algorithms can produce feasible paths for the UAV to travel, there is nothing preventing infeasible paths from being generated with respect to the UAV's flight mechanics. Current path planners generate a path to follow and then expect the UAV to follow the path to the best of its ability. This

presents a problem, because the UAV is incapable of performing the flight maneuvers required to follow the proposed path. For example, Figure 10 shows threats that the UAV should avoid with red circles. The dotted line is the planned path, but the solid line is the actual path taken by the UAV because of its flight constraints. This example shows the necessity of including flight capabilities in the path planning algorithm to guarantee a feasible path.

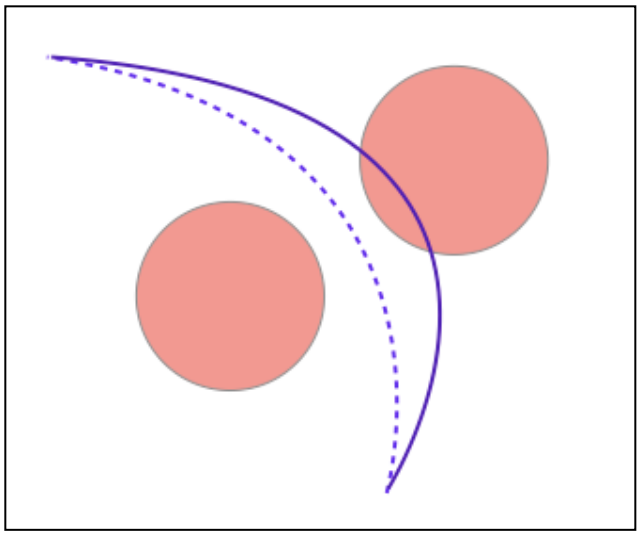

Figure 10: The impact of including flight dynamics in path planning

The second focus of the thesis will be on visualization of the information provided to the operator. Most interfaces for UAV command and control involve text displaying waypoints, speeds, headings, etc. While this works, it is not an intuitive interface that scales well for the model of a single operator controlling multiple UAVs. Previous work has been done on keeping the operator in the loop giving them a set of meta-paths to choose from and then the specific path. Meta-Paths are groups of paths with similar characteristics, such as fuel efficiency or threat avoidance [42]. This system keeps the number behind the scenes except for the total cost of the paths based on the optimization criteria. One drawback to the current system is linear representation of the UAVs flight path. Waypoints are generated and lines are drawn

between the waypoints to represent the flight path of the UAV. This representation allows for flight paths to be represented by 90-degree turns, something that is impossible. Visualizing the NURBS curve flight path allows operators to see how far off the linear path a UAV travels, particularly when making a turn.

Chapter Two details a literature review of current work in the areas relevant to UAV command and control as presented in this thesis. Chapter Three presents the methods used in modifying the path planning algorithm and interface. Chapter Four discusses the evaluation of the modifications compared to the previous work. Chapter Five reviews the improvements and discusses the possible impact on the system as a whole. Chapter Six outlines conclusions and future work on this topic.

Chapter 2 –Background

As discussed in the Introduction, a wide variety of fields must be brought together to generate a complete path planner system. The two major areas of path planning are the planning algorithm itself and the visualization of that information in a clear and concise manner. There has been much work on the command and control of UAVs, this section will briefly cover the areas relevant to integrating flight mechanics. Topics covered will be the Virtual Battlespace environment developed at Iowa State University, path planning methods and algorithms, flight mechanics, and path visualization.

Virtual Battlespace

This work is building off the Virtual Battlespace, seen in Figure 11, environment and path planning work done at Iowa State University [14][37][40][42]. The Virtual Battlespace environment was started in 2000 as collaboration between Iowa State University's Virtual Reality Applications Center (VRAC), the

Figure 11: Virtual Battlespace project

Air Force Research Lab's Human Effectiveness Directorate, and the Iowa National Guard's 133rd Air Control Squadron. The goal was to create an immersive virtual reality (VR) system for distributed mission training. The Virtual Battlespace environment takes information about tracks, sensors, targets, and threats and displays them to the operator in a VR environment that consolidates the information into a single coherent picture of the scene that can be viewed from multiple perspectives and scales [40][43]. The Virtual Battlespace environment displays scenario information graphically in order to reduce the amount of textual information the operator must read. In other research, a reduction of textual information through graphical representation was found to allow the operator more time to focus on mission critical decisions [44][45]. All interaction with the Virtual Battlespace environment is through a Logitech Cordless RumblePad 2 gamepad, Figure 12.

Figure 12: Logitech Cordless RumblePad 2

Path Planner was specifically designed to be viewed and manipulated in an immersive 3D environment. Each UAV path is defined by a series of points in 3D space (x, y, z position) and a destination time defining when the UAV will reach a specific waypoint. The UAV path is then visualized in the Virtual Battlespace

environment as a yellow fence, with each horizontal bar representing 1000 feet in altitude, as seen in Figure 13.

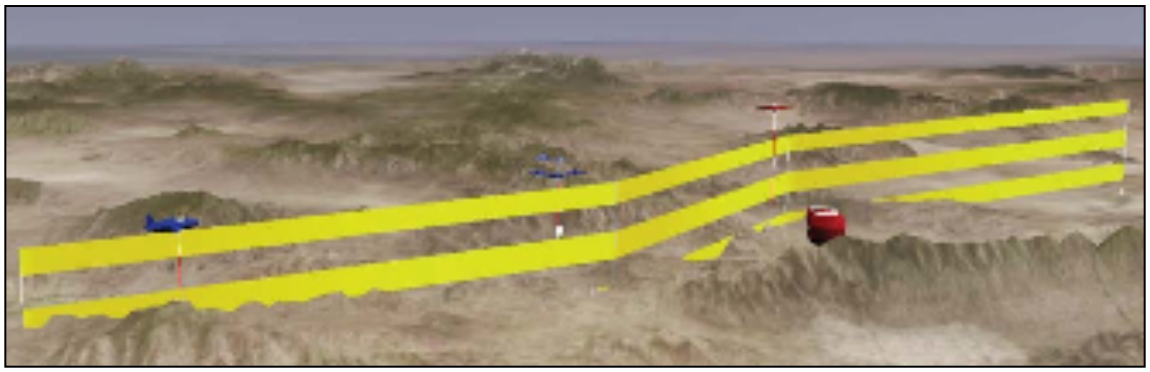

Figure 13: Virtual Battlespace path planner visualization of the UAV path

The waypoints are represented as bends in the fence and the orange boxes are gates the UAV will fly through along its path. Unlike traditional 2D visualizations of the UAV flight path, there are no artificial modifications to the waypoints because there is a one to one mapping of the waypoint location in three-dimensional space.

The Virtual Battlespace path planner generates multiple paths, by default fifteen paths, for the operator to chose from. The returned paths are optimized according to the criteria defined in Chapter Three. Operators can chose their best path from the group or they can chose to run path planner again.

Path Planning

Although path planning is a well researched area with many acceptable alternatives to creating a solution, there is currently no comprehensive solution to path planning. All solutions make sacrifices to solve part of the puzzle. Path planning can be done

in either 2D or 3D space. Online versus offline is also a large debate with each having benefits. Flight dynamics can require a large amount of computations, and as such are often left out of path planners even though there is a benefit from including such information in the calculation.

Two dimensional path planning algorithms are often used because they are simple to implement and the calculation speeds are faster than three dimensional. Many path planners accomplish this through allowing the UAV free motion in the horizontal direction but treat the vertical direction (altitude) as a constant [46]. Using this method allows ground path planning algorithms to be directly converted for use with UAVs. By reducing the degrees of freedom from 6 down to 4, the computational complexity is greatly reduced and allows for real-time path planning. 2D path planners lend themselves well to being visualized on a computer monitor because both have matching dimensionality (2D). 3D path planners require a form of simplification or dimensionality reduction to display the information on a 2D computer monitor. The drawback to this method is the gross simplification of the maneuvering ability of UAVs. By reducing the maneuverability of the UAVs, one of the greatest benefits of the UAV, flight, is ignored.

Three dimensional path planning algorithms are very attractive because they do not restrict the UAV's movements [47][48]. While moving to the full six degrees of freedom, allows for more accurate movement of the UAV, it also brings with it new problems to address. Two of the issues associated with moving from 2D to 3D are

the visualization of the environment and the computational time required to plan a path. The Virtual Battlespace environment can help with the visualization, because it is designed for stereo displays and immersive system allowing for 3D interaction with a scene opposed to the traditional 2D computer monitors. The computational problem with 3D can be addressed in multiple ways, from simplifying the calculations to parallelizing the computations. The benefits of 3D path planning over 2D provide motivation to solve these problems and make a more comprehensive and accurate path planning algorithm.

Offline path planners are used because they can provide more accurate paths than traditional online planners. Offline planners use all the situational information (UAV type, number of enemies, location of enemies, target locations, etc) to plan a path from take off to touch down before the UAV rolls onto the runway. Offline planners tend to be the most sophisticated algorithmically and the most accurate given certain constraints. They can be more sophisticated because there is no time constraint for generating a path. Offline planners can use complex vehicle dynamics models to ensure flight path feasibility [21][49]. These algorithms tend to favor accuracy over computational speed and as such are too slow to deal with dynamic situations where a UAV may need to re-plan the path in real time.

If dynamic re-planning were desired, an online path planner would be chosen. Online path planners focus on algorithms capable of running in real time. These algorithms favor computational speed over accuracy. Speed is important because a new path

must be computed before the UAV reaches the unforeseen obstacle that necessitated the re-planning. Constraints deemed less important, such as flight dynamics, are often sacrificed to achieve the real time speed desired. Some path planners update a specific section of the UAV's flight path that has become undesirable, such as discovering a threat covering part of the path. The path planner identifies the waypoints of the path that are no longer desirable and re-plan those specific waypoints points [50]. Others are constantly updating a short distance of the path in front of them to achieve constant feasibility, which is crucial when following a target [51]. This type of path planning requires fast computation because the frequency of which they are updating their path. Communication lag for UAVs can be several seconds, restricting this method to more autonomous vehicles. Both online and offline path planners make significant contributions, but the most elegant solution would be a balance between these features (e.g. providing as much accuracy as possible while remaining close to real time).

Vehicle dynamics are a critical piece of the path planning puzzle to make sure the paths returned are feasible for the UAV and prevents unintended drifting into undesirable areas. Offline path planners can take into account sophisticated vehicle dynamic models and simulations such as those modeled by Garza and Morelli [52]. The downside is the computational time necessary to calculate all possible drag coefficients for the UAV in varying winds. Some offline planners use only simple constraints such as the radius of curvature, velocity range, and altitude because

those are the most basic of UAV constraints [18][20][21][53]. Brintaki and Nikolos use a velocity range and altitude as constraints, but they modified their path planner formulation to use segment length and segment angle as design variables instead of x and y position [15]. Using this formulation they were able to constrain the turning radius of the path through the segment angle. They also claim better convergence replacing the traditional x and y waypoint location with segment length and segment angle. Offline path planners can use sophisticated flight dynamics, like Garza and Morelli, in their constraints because they are not limited in computational time. Instead offline planners often choose to simplify the vehicle dynamics to something reasonably accurate and to focus on the optimization formulation and method used.

Online path planners for aerial vehicles sometimes use the same simplified version of vehicle dynamics derived from ground based vehicle dynamics. These algorithms assume the UAV flies only in a horizontal plane and do not consider climbing or descending flight paths [21][27]. By eliminating the third dimension, the algorithms are greatly simplified and their computational speed is thus very high. However, eliminating the third dimension also reduces the operability of the UAV and eliminates one of their strongest advantages over ground vehicles. Consider a UAV operating close to the ground where terrain can become an issue. If the UAV is traveling toward a ridge, the best path could be to fly vertically a few hundred feet to fly above the ridge and continue on the original path. By keeping the altitude

constant, the UAV may have to travel many miles from the original path to find a way around the ridge.

A complete vehicle dynamics model of the UAV would provide the most realistic results but such calculations require greater computing power. However, there are some basic constraints that should always be taken into consideration when planning a path, such as the minimum turning radius, velocity range, altitude range, and the maximum load factor (force on the plane). These constraints allow accurate representations of UAV behavior with limited computational effort.

One way to plan paths is to set objectives/criteria for the UAV to achieve in the cost function formulation. Objectives could be things like path length, avoiding threats, and following certain objects. Other objectives could focus on flight dynamics, such as minimum radius of curvature, range of velocities (stall to maximum), or climbing/gliding rates. Most path planners do not include flight dynamics in the cost function formulation, instead assuming the UAV will travel as close to the stated path as possible. When using optimization algorithms to generate paths, unconstrained problems are often used because they are easier to code and solve. Constraints, such as flight dynamics, can be treated as components of the cost function. By setting the cost of violating these constraints prohibitively high, the possibility of an optimal path breaking one of the constraints is effectively eliminated.

There are many cost components used across the various forms of path planners. What is included in the cost function is not based on a standard but on the

programmer and their goals for the path planner. One example of a cost component is radar cross section, to improve the stealth of the path and decrease the likely hood of being shot down [17]. While path planners may have similar objectives, the cost component formulation may be very different and as such generate vastly different paths. There is still much work needed to find the best mathematical formulations of cost functions to achieve a desired path.

Flight Dynamics and Mechanics

Flight dynamics is the vehicle orientation of air and space vehicles in three directions. The angle of pitch, roll, and yaw about the vehicle's center of mass are the basis for most flight dynamics and can be seen in Figure 14. These three dimensions are critical to defining the flight path and forces associated with the vehicle. The four basic design cases for flight mechanics are, straight and level flight, turning, approach and landing, and takeoff. While these are the four most basic design cases, there are many ways to approach them with varying levels of sophistication.

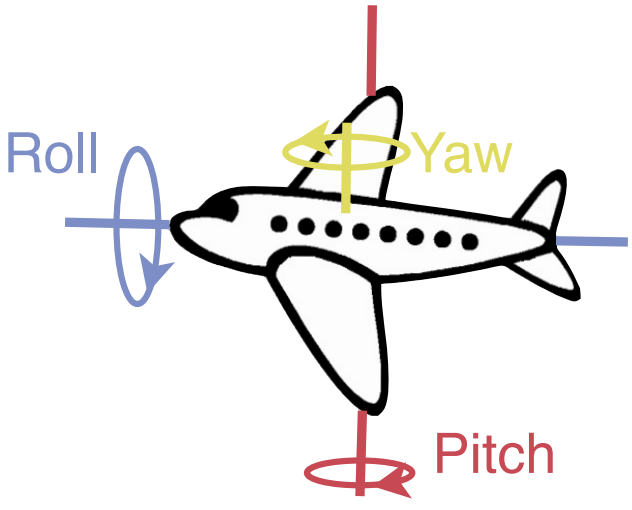

Figure 14: Pitch, Roll, and Yaw of an aircraft

The most basic mechanics of flight considers the forces on an aircraft in equilibrium conditions and the responses of the aircraft to those forces. These equations

are generic across all types of aircraft and involve the most basic forces of Lift, Weight, Thrust, and Drag. The goal is to balance the forces on the aircraft, so the pairs Lift/Weight and Thrust/Drag are always equal. The forces acting on an aircraft are shown in Figure 15 and show the pairing of forces [54][55]. Velocity is also important in calculating the forces when the aircraft is turning or pulling out of a dive. While these equations give a basic understanding of the forces acting on the aircraft, they do not take into consideration forces on the individual airfoils, aerodynamics, or systems not in equilibrium.

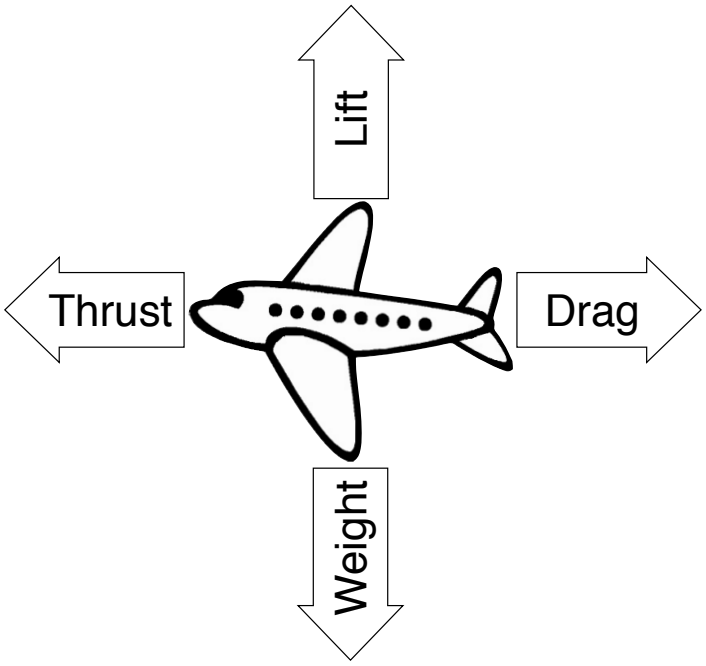

Figure 15: Balanced forces on an aircraft

More complex flight dynamics models and simulations are all built upon the basics, but start to incorporate more information and complexity. Dynamics models typically use one of two methods to describe the aircraft, either using Euler angle formulas (pitch, roll, yaw) or more computationally efficient quaternions [56]. These methods allow the representation of an aircraft in all six degrees of freedom. Incorporating atmospheric conditions, aircraft characteristics, and other constants such as Lift or

Drag constants, provides a complete model with higher accuracy than the basic models [57]. The more complex the models are the more accurate representations of the aircraft flight but also the more computational power required. The trade-off between flight accuracy and computational efficiency becomes a serious concern, how accurate should the flight dynamics be to be effective enough in path planning? Flight simulators were some of the first software packages focusing on providing realism through the use of flight dynamics. Flight dynamics can be programmed using very sophisticated algorithms in different languages, for example MatLab, coordinating information specific to the aircraft to the model's behavior [52]. The focus of simulators is primarily the training of pilots, so the more realistic the environment, the better the training by replicating the nuances associated with flight. By using non-dimensional coefficients for longitudinal, lateral, and control coefficients for the forces exerted on the aircraft, the computational efficiency can be saved while providing a more realistic training environment [26]. Flight simulators need to have sophisticated algorithms for simulating flight, but UAV path planning does not require capturing all the nuances.

Path planning generally uses basic characteristics of flight dynamics without incorporating a complete model. The most common flight characteristic implemented in path planning is the minimum turning radius of the aircraft. This is because a minimum turning radius can have the greatest impact on returning feasible paths. Without a minimum turning radius, planes would be able to make 180-degree turns

instantaneously to follow a path. The basic mechanics of flight equations state that the minimum turning radius of a aircraft is dependent on the speed the aircraft is turning, assuming no slipping [55]. Most path planning algorithms do not consider velocity in their minimum turning radius in their calculations, instead setting a hard value within reason [20][21]. The banking angle of the aircraft is another flight path characteristic that holds value for determining radar cross section and as such, the path's ability to keep the aircraft undetected [17].

All aircraft have a minimum velocity they must maintain before stalling, which is an important flight dynamic to consider when planning a path. Similar to the stall velocity, all aircraft have a ceiling, or altitude they cannot climb above. Taking into account the ceiling is vital to planning a safe path. Path planner algorithms have started integrating some important basics but still lack a few critical constraints. Along with a stall velocity and ceiling, all aircraft have a maximum load they can experience before structural failure. A combination of the minimum turning radius and stall velocity is how the effective stall velocity of an aircraft changes with the speed and radius of a turn. Using equations for minimum turning radius and effective stall velocity the flight path can be evaluated for feasibility for a given aircraft.

B-splines

UAV paths are often represented using a b-spline curve, because it represents a smooth transition between waypoints. There are two ways to represent a b-spline curve, using the waypoints as control points and generating a curve, or interpolating

new control points so the curve passes through the original waypoints. The traditional way to define a b-spline curve is through control points, a knot vector, and a degree. All three components are used to generate points on a curve. The downside for path planning is that the curve only touches the end points of the curve, and not any of the other waypoints as shown in Figure 16. Inaccurate paths are created when the waypoints are used as control points of the b-spline curve.

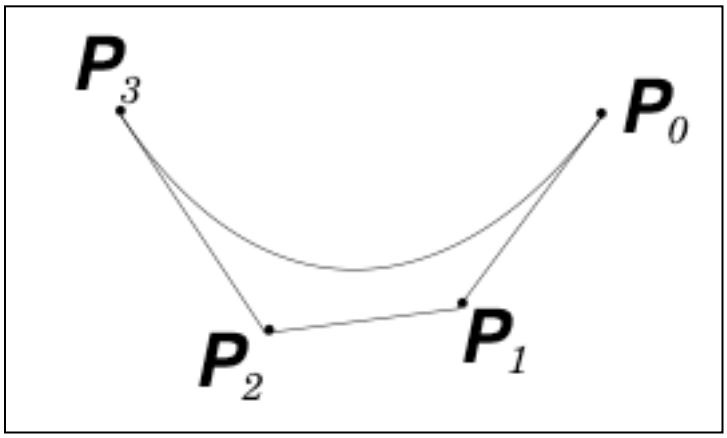

Figure 16: B-spline representation with control points (P)

Equation 7 describes pth-degree b-spline curve where P_i are the control points and $N_{i,p}(u)$ are the pth-degree b-spline basis functions defined on the non-periodic knot vector (**U**) from 0 to 1. The basis functions can be defined in a number of ways, divided differences of truncated power functions, blossoming, and a recurrence formula. The recurrence formula is the most useful for computer implementation and what is described in Equations 8 and 9. U is the knot vector, u_i are the knots, and the basis functions are defined by $N_{i,p}$ where N is the ith b-spline basis function of p-degree [38].

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \tag{7}$$

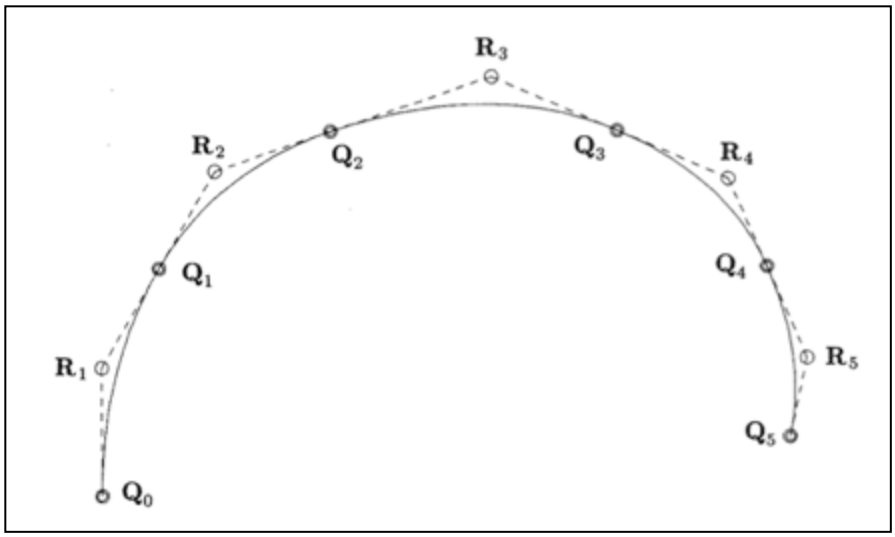

Figure 17: Interpolated B-Spline representation

$$N_{i,0}(u) = \begin{cases} 1 & if\ u_i \leq u\ \leq\ u_{i+1} \\ 0 & otherwise \end{cases} \tag{8}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1} + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{9}$$

The other option is to interpolate a new set of control points that generate a path, which passes through the original waypoints. The Q points in Figure 17 are the original control points for the b-spline, and the R points are the new interpolated control points to generate the solid b-spline curve traveling through the original Q control points. There are multiple techniques used for interpolating a b-spline using different parameterizations such as uniform length, chord length approximation, centripetal, and universal [38][58][59]. All involve different ways to interpolate new control points, with some more prevalent to having the curve wrap around itself or different ways to interpolate the derivatives at the end points. Uniform length, chord

length approximation, and centripetal interpolation determine the parameters of the curve before defining a knot vector. Universal parameterization generates a uniformly spaced knot vector and computes the parameters from the knot vector. Chapter 3 will cover universal parameterization in more depth, including equations and how it was utilized in the Virtual Battlespace Project.

Path Visualization

Visualization is a critical aspect of path panning that is often overlooked to place more focus on the algorithms used to plan the optimal path. Path visualization is the connection between operator and vehicle, and as such is vital to successful navigation in hazardous areas. Visualization will become more important as the military continues pressing for a single operator to control multiple UAVs. Effective visualization methods allow the operators to maintain awareness of all UAVs while controlling one specific vehicle. Most path planners use 2D interfaces for displaying UAV paths, which limits the true versatility of the UAVs.

Currently, most path planner interfaces focus on displaying information in 2D. This makes sense when the majority of the path planners hold the altitude constant, effectively eliminating the third dimension [60]. Some interfaces take the third dimension and display them in a second 2D window [61]. Figure 18 is an example of this split screen technique. The top graph is a top down view of the environment and displays the vehicle's heading and path. The bottom graph illustrates the altitude of the vehicle at any given time during the path. While this technique presents the third

dimension, it can be confusing to relate the vehicle's position from the top graph with the altitude in the bottom graph.

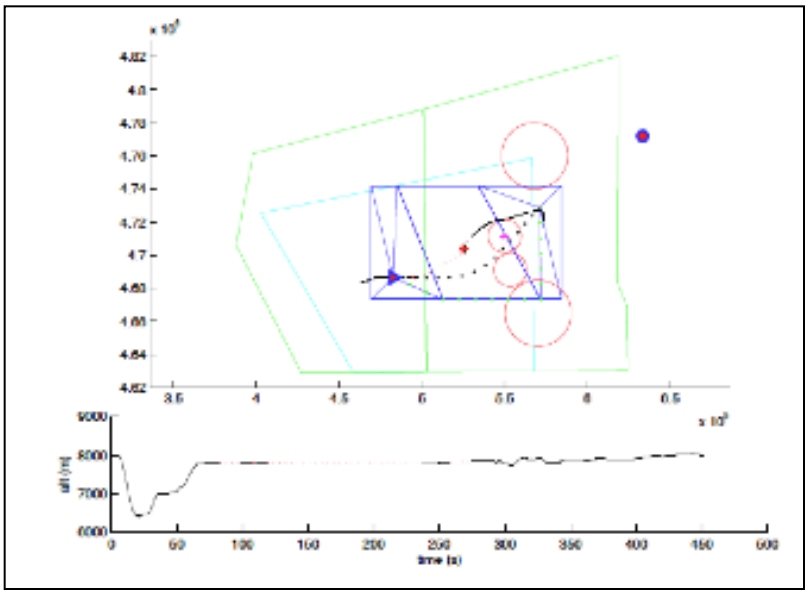

Figure 18: 2D path planner displaying altitude

A common way to do 3D visualization of path planning involves establishing a single isometric view of the engagement zone [62]. Figure 19 is the isometric view type interface. This view gives the operator a better idea of height, specifically in relationship to the terrain. The drawback to this interface is the lack of interaction with the view. Once the isometric view is set, the operator cannot rotate views or interact with the scene to get a better view.

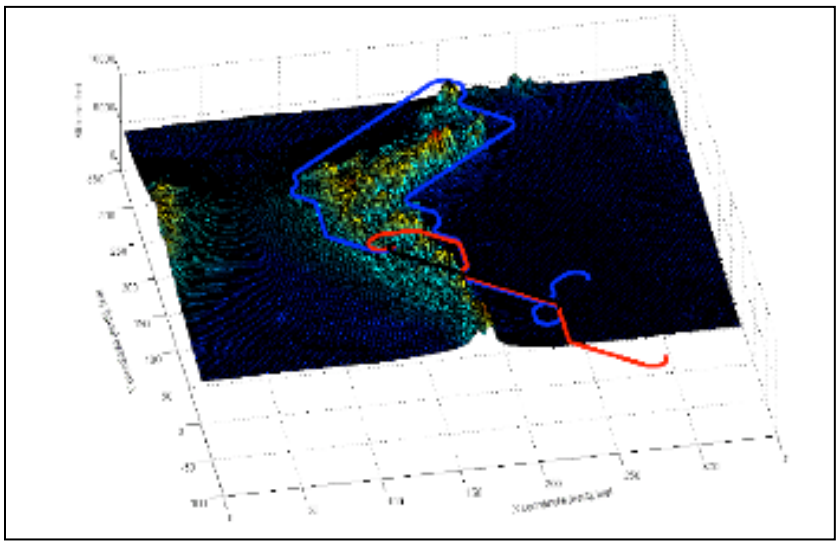

Figure 19: 3D path planner showing isometric view

All the previous interfaces display a single optimized path that the vehicle will take. Path planners are

not stable enough to completely rely on for planning missions autonomously. This requires a human to be in the loop for decision-making. A human operator wants to see multiple optimal paths to choose the best choice, making the path visualization more important than in fully autonomous path planners. Unfortunately, while visualization is more important for human in the loop path planners, the visualization techniques are still primitive. Figure 20 shows a path planner for dynamic environments that allows the operator to select between two alternate paths [19]. Another alternate is to give the operator varying degrees of aggressiveness for the alternate plans to choose from for each obstacle [22].

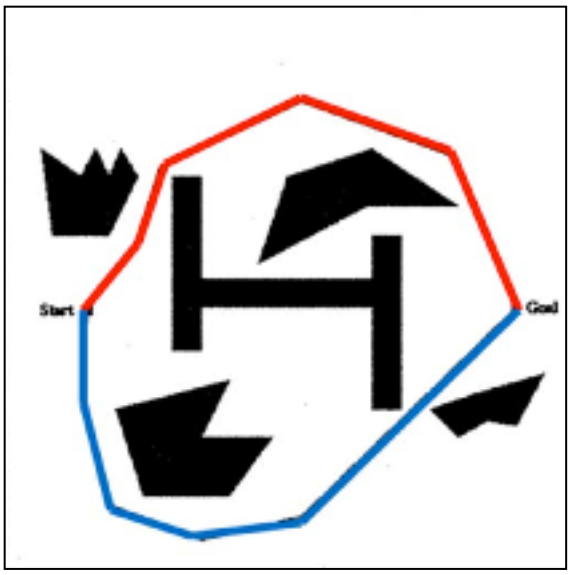

Figure 20: 2D path planner showing alternate paths

One 3D approach to visualization of alternate paths is to group paths with similar characteristics, such as fuel efficiency, threat avoidance, or reconnaissance, into a single “meta-path.” By grouping paths with similar characteristics, the operator can choose what type of path they would like to choose, e.g. a fuel efficient path, and then select the specific path. Figure 21 shows “meta-paths” and their use in a 3D environment, with each surface color representing a different group of path characteristics [42]. The drawback to this interface is the use

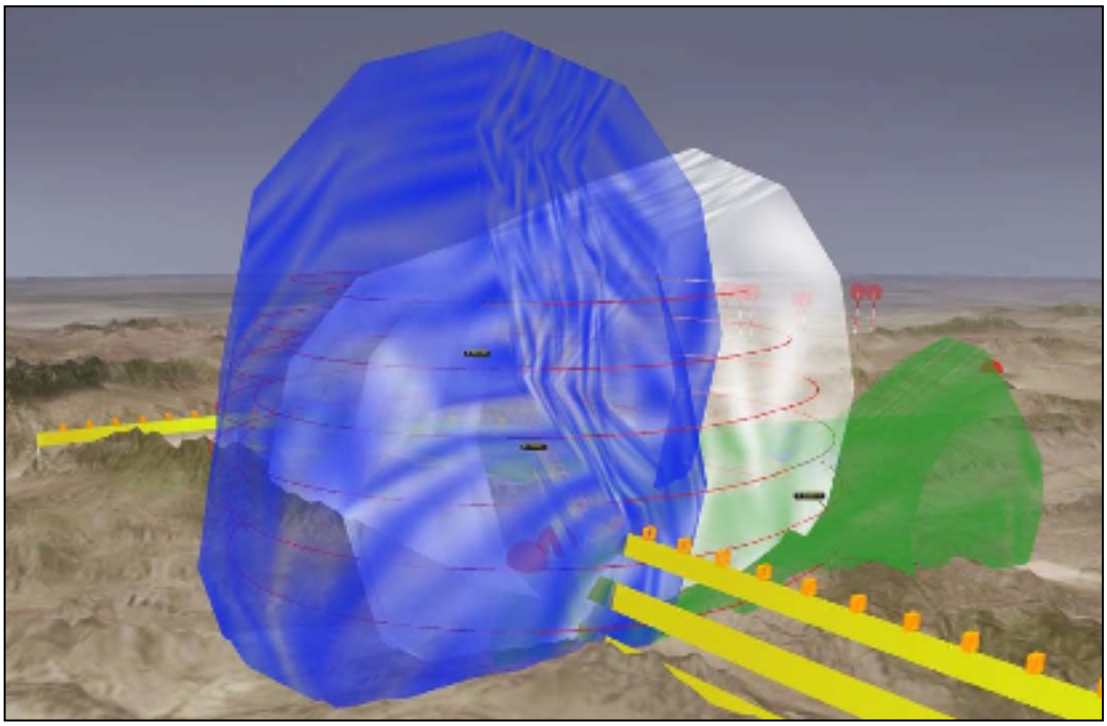

Figure 21: Virtual Battlespace meta-paths visualization

of linear interpolation between UAV waypoints. This method does not show the curved path taken by a UAV to make turns or pull out of a dive.

Non-Uniform Rational b-splines or NURBS are well accepted representations for surface modeling that have been used in the CAD and Modeling industries for years. NURBS are the preferred method of representing curves and surface information because surfaces can be generated using few control points and because they are weighted, they can represent any geometry exactly. Visualizing the flight path as a b-spline curve would provide the operator with a more realistic flight path taken by the UAV through the specified waypoints.

Summary of Path Planner Research

Path planners cover a wide range of research issues from optimization to visualization. The main focus of path planners has been on the optimization algorithm. Current path planners use information such as threat locations and terrain information to plan paths, but they do not consider basic flight mechanics to ensure feasible paths. Flight simulators have been using sophisticated dynamics models for planes for many years. These dynamics models are very accurate at capturing the nuances of the plane's flight mechanics.

Most path planning algorithms have not yet started integrating flight dynamics into their calculations because of the computational resources required. Most offline path planners do not integrate advanced flight dynamics, even with the added computational time afforded them, because the flight models are overly complex for path planning. The most advanced flight dynamics implemented in online path planners are constants, such as a minimum turning radius, a maximum altitude, and a speed range that can be traveled by an aircraft. While these characteristics are helpful in generating a more feasible path, things like the minimum turning radius and the effective stall velocity change as the aircraft moves.

The next issue comes in visualizing the flight path of the aircraft. Currently, 2D visualizations of the engagement zone are the preferred method, with the altitude of the aircraft being held constant. Some 3D path planners are using multiple viewing positions to allow the operator to see the scene in 3D. 3D viewing helps immerse the

operator in the scene because the aircraft is no longer constrained to a single altitude, but can move in the virtual world the same way it can move in the real world. Other visualization issues come into play when trying to convey dynamics information to the operator.

Research Issues

Based on the conclusions drawn from the literature review, two research issues have been identified:

1. *Improve the cost function by incorporating flight mechanics to provide feasible paths.*

   The optimization cost function was improved through the incorporation of flight mechanics to provide a more accurate representation of the UAV's capabilities. Basic flight mechanics equations relating to the force exerted on the aircraft, the effective stall velocity, altitude constraints, and the minimum turning radius as a function of speed were used to provide feasible paths in close to real time.

2. *Develop a visualization strategy to more accurately display the aircraft's flight path.*

   Having the human in the loop requires providing the operator with visualizations to keep situational awareness at all times. Creating a more accurate representation of the flight path using b-spline curves will allow operators to react quickly and analyze paths with greater efficiency.

Chapter 3 – Method Development

Previous Work

Original Algorithm

The general approach for path planning is defined in Figure 22. The base algorithm was modified to enhance the fidelity of the simulation and to explore other visualization methods.

These enhancements are the work presented in this thesis, not the original base algorithm. This work builds upon the platform of the Virtual Battlespace environment for UAV path planning and battlefield command and control.

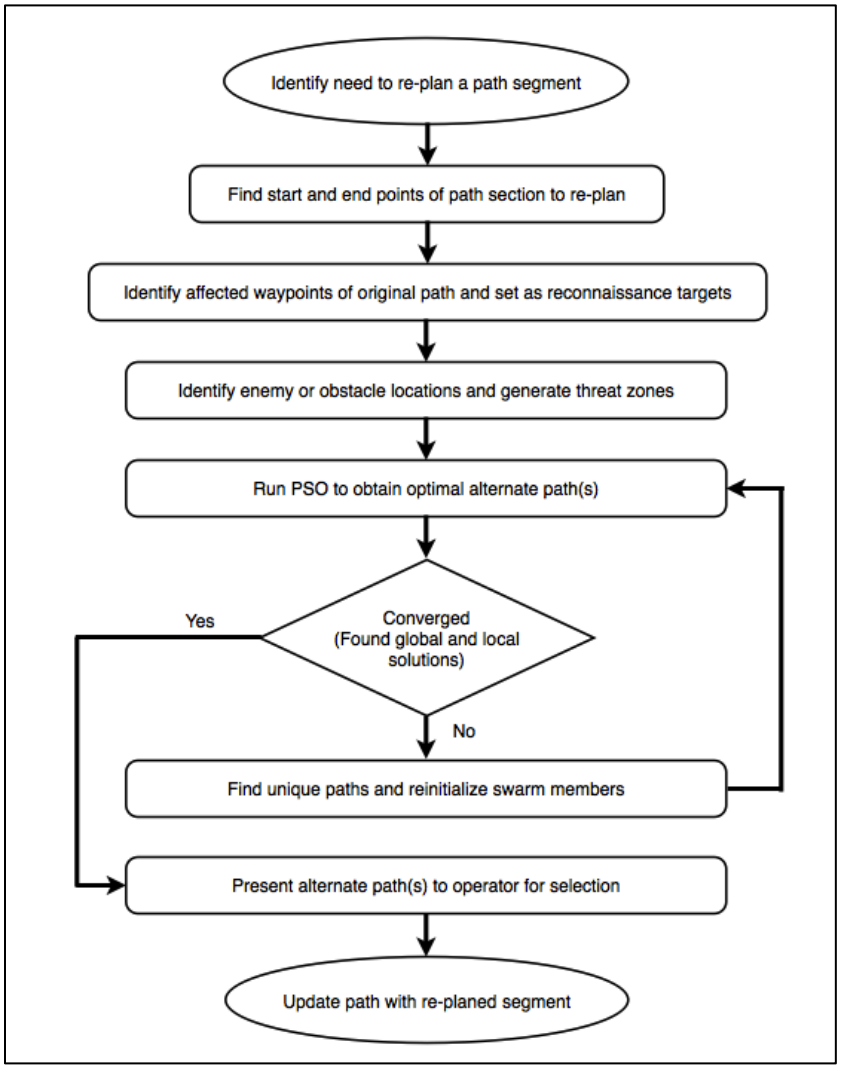

Figure 22: Virtual Battlespace's path planning flowchart

The path planner starts when a UAV identifies something in the environment that requires a new path, such as a SAM site threat. When the UAV determines a new path is needed, Virtual Battlespace prompts the operator with an alert. The alert is designed to allow an operator to control multiple UAVs in the same environment while ensuring any events requiring operator intervention do not go unnoticed. The operator then determines whether they wish to ignore the alert and have the UAV continue on course or investigate the alert for a new path. Once the operator determines an investigation is necessary, they select to move into Virtual Battlespace's path planning mode.

The first step in the path planning process is to determine the starting and ending locations for the alternate path segment. This is done by projecting the location of the UAV a set time in the future from the time the user decides to investigate the alert (by default 30 seconds). The lead-time is the amount of time the operator has to run the path planner and choose a new path before the UAV encounters a threat. The ending location is determined by simply stepping along the original path until a location is reached that falls outside of a threat. This location is deemed safe and selected as the end point for the new path segment.

Next, the waypoints along the original path segment being re-planned must be set as reconnaissance targets. This makes sure the path planner will plan a new route close to the original path, so any reconnaissance targets the UAV was originally tasked to investigate, defined by the waypoints, will still be investigated. The path

planner must now determine threats to avoid with the new path. Entities close to the original path are evaluated and classified as either a threat or an ally. Both enemies and allies must be avoided, enemies because they are threats and allies because of potential crash hazards.

All of this information, start point, end point, reconnaissance targets, and threat locations, are then fed to the PSO path planner. The PSO algorithm takes the data and generates alternate paths based on user-defined objectives and returns the results in the form of visualized paths in the Virtual Battlespace environment. Path generation and visualization are decoupled allowing research on both the optimization algorithm and the visualization methods independently. Work on the optimization algorithm will be presented first, followed by the visualization techniques.

Figure 23 shows a simplified 2D illustration of the 3D threat avoidance problem. The green line represents the original UAV path. The big red circle represents a threat zone (also referred to as threat domes). The black dot in the middle of the red circle is the threat location (Z_T), for example

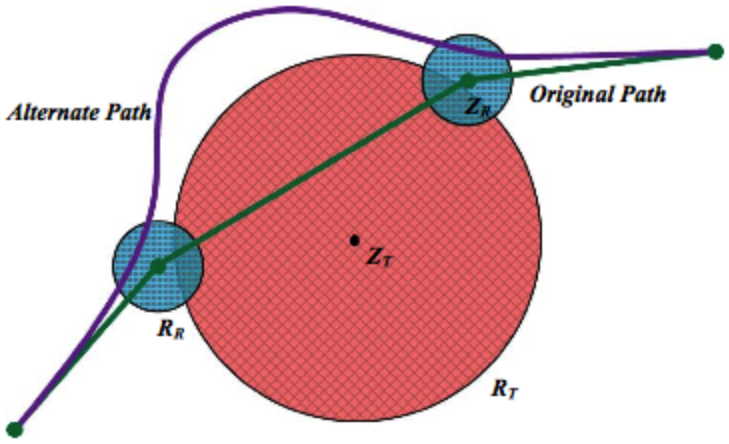

Figure 23: Threat avoidance represented in 2D

an enemy SAM site. The two smaller blue circles represent the reconnaissance zones the UAV must fly through to complete its mission, with locations centered at Z_R. The curved purple line is the alternate path that avoids the threat zones and passes through the reconnaissance zones.

The path planning process will begin because the green path passes through an expected threat zone, triggering an alert. The operator must then select to investigate the alert, initializing the path planning process. The green endpoints are defined as the start and stop points for the new path segment with the original path points between those two points acting as reconnaissance points (Z_R). From these points, the path planner will initialize a 3D design space to search for an optimal path. The design space must be properly sized to allow efficient operation of the PSO algorithm. A design space too small would limit the chances of finding a truly optimal solution, while a design space too large would increase the computational time to finding an optimal solution.

This simplified 2D version of the 3D path planner problem, illustrates the goal of finding a path that can avoid threat zones while still traveling through the reconnaissance zones. The purple line is a representation of a solution the path planner might generate to meet these objectives. PSO was chosen because of its ability to find a compromise between objectives as it searches for an optimal path. For example, PSO can rank more fuel efficient paths as more optimal if the user determines that objective to be more important than reconnaissance or safety.

Original Particle Swarm Optimization Problem Formulation

Particle Swarm Optimization (PSO) emulates the actions of bugs swarming around food in a mathematical algorithm, with the food representing optimal designs. PSO begins by randomly initializing swarm members throughout the design space. In this case, the design space consists of all possible combinations of waypoint positions (x, y, z coordinates) to make a single path within the given constraints. The waypoint locations are the design variables. Each member of the swarm represents a single path generated by the path planner, i.e. a set of waypoints for the UAV to follow. Each path is modeled using b-splines to generate smooth curves between waypoints. The size of the swarm is the number of paths computed in parallel at a given time. The swarm size is user defined, with the tradeoffs of more swarm members requiring more calculations per cycle versus a smaller swarm size being calculated faster each cycle, but may take more time to completely explore the design space, if at all. The optimal path is calculated as the smallest cost function value of all swarm members.

Paths with the lower computed total cost, are considered more optimal than paths with a higher total cost in this planner. The total cost of the path is determined by a set of cost functions to quantify the path's ability to meet certain criteria, such as fuel efficiency, threat avoidance, reconnaissance, or flight mechanics. Calculating the total cost requires dividing the b-spline flight path into segments. Two points, a start and end point, with their own x, y, and z positions, represent each segment. The paths can be broken into any desired number of segments, but there is a tradeoff,

with more segments the accuracy of the path increases, but the computational time also increases.

Once the path has been broken into segments, we can evaluate the path's cost value to determine where it ranks in relationship to the other swarm members. The cost value is a calculated number signifying the ability of the individual path to meet the user specified objectives, such as threat avoidance. The optimization cost function is a combination of all user defined objectives expressed in mathematical form. Four objectives were considered in the original formulation of this path planner, fuel efficiency (C_F), reconnaissance (C_R), threat avoidance (C_T), and terrain collision avoidance (T). The combination of these objectives into a single optimization cost function is:

$$C = K_1 C_F + K_2 C_R + K_3 C_T + T \tag{10}$$

K_1, K_2, K_3, in Equation 10 are constants referred to as the component weights, which determine the relative emphasis each objective has on the total cost of each path. The component weights can be changed or shifted by the operator based on personal preference or mission specific requirements, for example placing more emphasis on reconnaissance to make sure the UAV sees the target even if the vehicle might be at a greater risk for being shot down. Component weights are set between 0 and 1, with a component sum of 1 when all components are added together. PSO finds the optimal path solution by adjusting the design variable, the waypoints, until the total cost of the path is the lowest it can return for the design

space. The original cost function formulation [14] and subsequent modifications [37] will be discussed.

Re-initialize Swarm Members

One drawback to the traditional PSO path planner method is that the top few paths returned by the algorithm will be very similar. For most cases of PSO this is desired to find the global best solution. However with path planning, there can be a variety of solutions that could produce a similarly acceptable solution. The problem is then how to tell the algorithm to generate a more diverse set of solution paths. An approach is to tell path planner to re-initialize swarm members that are similar enough (as defined below) to be considered the same basic path. To do this, path planner checks all paths against the *gBest* and the paths that are functionally identical are re-initialized. This prevents the algorithm from returning all similar paths and it forces the exploration of the entire design space.

Equation 11 is the first step in the re-initialization process, where the swarm member n, ($fitness_n$) is evaluated against the global best, gBest fitness ($fitness_{gBest}$). If the difference is less than the user defined value C_V (for this algorithm $C_V = 0.99$) then the path is considered unique and the swarm member remains unchanged. If the difference is greater than C_V the path needs to be evaluated further to determine if the path is unique based on the 3D position, as shown in Equations 12 and 13. To measure the distance between the 3D positions in space, the algorithm steps through the segments of both the path n and the global best gBest. The square of

the distance between segment *i* on path *n* at location [$n(x_i)$, $n(y_i)$, $n(z_i)$] in 3D space and the corresponding segment *i* on the global best, *gBest*, is denoted by p_n (also in 3D space) is computed and summed over all *N* path segments and denoted at D_N. If the separation, D_N, is less than the user defined tolerance, C_l, the paths n and gBest are considered functionally identical and the swarm member is reinitialized. If the separation, D_N, is greater than the user defined tolerance, C_l, then nothing is done to the swarm member.

$$V = \begin{cases} true & \left|\frac{fitness_n}{fitness_{gBest}}\right| < C_V \\ false & \left|\frac{fitness_n}{fitness_{gBest}}\right| \geq C_V \end{cases} \tag{11}$$

$$D_n = \sum_{i=1}^{N} \left(p_{gBest}(x_i) - p_n(x_i)\right)^2 + \left(p_{gBest}(y_i) - p_n(y_i)\right)^2 + \left(p_{gBest}(z_i) - p_n(z_i)\right)^2 \tag{12}$$

$$I = \begin{cases} true & D_n < C_I \\ false & D_n \geq C_I \end{cases} \tag{13}$$

Fuel Efficiency Cost Component

Fuel efficiency cost component, C_F, is currently determined solely using the length of the UAV path and does not take into consideration any flight dynamics or environmental constraints. Estimating fuel efficiency based only on the path length is a rough approximation. The fuel efficiency cost component, C_F, is defined as the extra distance traveled if the UAV takes the alternate path compared to the original. Equation 14 shows how to sum up the segments of both the original path and the

proposed path. The path length, *L*, is the sum of path n's line segments $p(u_i)$. L_0 is the sum of all line segments comprising the original path, thus the distance the UAV would have traveled before re-tasking. To eliminate the possibility of having a negative fuel efficiency cost component as well as balance the fuel efficiency cost with the other cost functions, the cost was made linear with a minimum value of 10.

$$C_F = 10\left(\frac{L}{L_0}\right), L = \sum_{i=1}^{N-1} |p(u_{i+1}) - p(u_i)| \quad (14)$$

Reconnaissance Cost Component

The second cost component is reconnaissance. It is assumed that the initial path's waypoints are ideal reconnaissance and the areas around the waypoints are acceptable, but the reconnaissance degrades the further the UAV gets from the original waypoints. It is also accepted that reconnaissance is possible from farther locations, as long as the UAV has a clear line of sight and viewing angle. Figure 24 shows how the viewing angle and distance relate to a UAV's ability to perform reconnaissance.

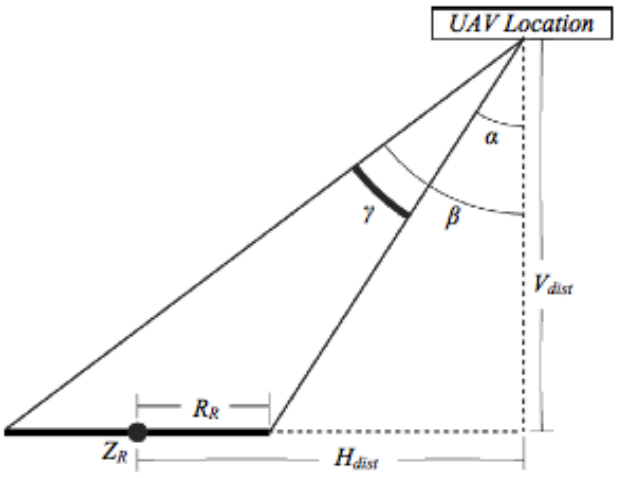

Figure 24: Reconnaissance diagram

The reconnaissance cost component is defined by two

values, the distance from the target, and the direction. Equation 15 calculates the distance of the UAV from each segment of the path $p(u_i)$ and the reconnaissance zone, Z_R. The distance is then scaled by the reconnaissance target's radius, R_R. The second component, the direction is shown in Equation 16. For each reconnaissance target the minimum distance and direction for any path segment $p(u_i)$ is found, shown in Equation 17. The summed minimum value is scaled by ten over the number of reconnaissance targets N_R, as shown in Equation 18, to produce the reconnaissance cost component.

$$d(p(u_i), Z_R) = \frac{Distance\ between\ points\ p(u_i)\ and\ Z_R}{R_R} \tag{15}$$

$$\theta_i = \tan^{-1}\left(\frac{H_{dist,i}}{V_{dist,i}}\right) \tag{16}$$

$$C_j = min(\theta_i + d(p(u_i), Z_R), i = 1,2, \dots, N), \qquad j = 1,2, \dots, N_T \tag{17}$$

$$C_R = \frac{10}{N_R}\sum_{j=1}^{N_R} C_j \tag{18}$$

Threat Avoidance Cost Component

The third cost component of the path planner is threat avoidance. To determine the threat avoidance cost, the distance between the UAV and the threat must be determined to identify if the UAV has breached the threat zone, Equation 19. The

distance between UAV and threat is D_{Tj}, and the radius of the threat zone is R_T. To reflect threat zones being more dangerous the closer to the center, a non-linear approximation was used, Equation 20. The cost, $C_{i,j}$, is evaluated for each path segment. All segment cost values are then summed and scaled by the number of threat zones. If the UAV never breaches a threat zone the segment cost values would all be 0, but the total threat avoidance cost component, C_T, would be 10 because of the added constant. This is done to ensure the threat avoidance cost component is the same value as the rest of the cost components.

$$d(p(u_i), Z_T) = (R_T) - (Distance\ between\ point\ p(u_i)\ and\ threat\ Z_T) \tag{19}$$

$$C_T = 10 + \left[\frac{1}{N_T}\right]\sum_{i=1}^{N}\sum_{j=1}^{N_T} C_{i,j}\ , \quad C_{i,j} = \begin{cases} 1 - \dfrac{\sqrt{D_{T,j}}}{\sqrt{R_T}} & D_{T,j} < R_T \\ 0 & D_{T,j} \geq R_T \end{cases} \tag{20}$$

Terrain Collision Avoidance Cost Component

Terrain collision avoidance must be considered to insure paths do not travel through the terrain. The large cost, approximately 10 times that of the other three components, associated with violating the terrain boundary makes returning paths that violate the terrain boundary unlikely. Determining the terrain collision avoidance cost begins with determining the distance (altitude) between the UAV and the defined floor, in this case 500 meters. Once distances have been calculated Equation 21 shows the resulting cost. If the distance is greater than the minimum safe distance, in this case 500, there is no cost associated. If the distance indicates

the UAV has breached the terrain boundary, a cost of 1000 is added to the cost formulation.

$$T = \sum_{i=1}^{N} T_i \ , \quad T_i = \begin{cases} 1000, & h_i \leq 0 \\ 0, & h_i > 500 \end{cases} \tag{21}$$

B-splines

To evaluate the path's of UAVs there must first be a set of positions (path) to evaluate. The UAV's path is represented by Non-Uniform Rational B-splines (NURBS). NURBS have become the desired tool for representing geometric information in computer processing. They allow all types of geometry to be represented, whether curves or surfaces, and can be stored in small data sets comprised of control points ($P_i(u)$), degree (k – 1), and a knot vector (**U**). Points on the b-spline curve are defined by n+1 control points, the degree (k - 1), and a set of recursively generated Berstein basis functions ($N_{i,k}(u)$) . Equation 22 shows the relationship between the control points, degree, and basis functions. The Berstein basis functions are generated using Equations 23 and 24.

$$p(u) = \sum_{i=0}^{n} P_i N_{i,k}(u) \tag{22}$$

$$N_{i,1}(u) = \begin{cases} 1 & if(x_i \leq u \leq x_{i+1}) \\ 0 & otherwise \end{cases} \tag{23}$$

$$N_{i,k}(u) = \frac{(u - x)N_{i,k-1}(u)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - u)N_{i+1,k-1}(u)}{x_{i+k} - x_{i+1}} \quad (24)$$

While b-splines have many characteristics we find desirable in a path, the one weakness in terms of path planning is that the curve does not pass through the control points. This is not a problem for most types of geometric applications, but can be a problem for UAV path planning. For example, waypoints define where the UAV must go. If waypoints were used to generate the UAV's curved path, the paths generated rarely are one that travels through the waypoints. To alleviate this problem b-spline interpolation can be used. B-spline interpolation can be called b-spline curve fitting because it attempts to fit a curve to the set of control points, in this case waypoints, given. This process is mostly reverse engineering, where a curve is approximated to fit the data given and generating a new set of control points for that curve. The ideal interpolation would provide curves that mimic the original shape of the vertex-based model accurately, have minimal distortions, and be stable enough to accommodate higher order polynomials.

Visualizing flight paths to operators is important for efficient decision making. Visualizing UAV flight paths using b-splines provides the benefits of a smooth, accurate, and realistic representation of the path. Using b-spline interpolation allows for flight paths that travel through the desired waypoints, improving on the current linear interpolation method currently employed.

There are many different methods used to interpolate b-spline curves, uniformly spaced, chord length approximation, centripetal, and universal, but all use the same four steps. Calculate a knot vector and select the parameter “u”, the normalized parameter ranging between 0 and 1. Calculate the Basis Matrix. Solve for control points. Perform b-spline approximation. The difference in the interpolation methods is in the first step of calculating the appropriate parametric “u” values.

The universal method of b-spline interpolation was selected for this application because of the stability provided. This method is known to produce small wiggles when there is an adjacent long chord. However, long chords do not have bulges, as seen in other interpolations like chord length. The advantages of universal interpolation outweigh the disadvantages because UAV paths are generally composed of long chords [58].

This project focuses on utilizing flight aspects of the UAVs to allow the path planner to return flyable paths to the user. To calculate these flight mechanics, the position and velocity of the UAV at a given moment along the flight path are evaluated. Velocity is the derivative of a point on the curve, which can be calculated using recursive b-spline algorithms.

Equation 25 represents the calculation of the kth derivative of C(u) given by $C^k(u)$ [38]. For any point on the curve (u) the derivative ($C^k(u)$) can be computed using the kth derivative basis functions and a set of control points (P_i). This allows us to

calculate the instantaneous velocity, magnitude and direction, of the UAV at any point along the path.

$$C^k(u) = \sum_{i=0}^{n} N_{i,p}^k(u) P_i \tag{25}$$

Formulation of Vehicle Mechanics Equations

Integrating vehicle mechanics into the path planning problem formulation is imperative to making sure the planned paths are feasible for flight for the UAVs. There are many flight models available that model the flight characteristics of individual planes. These models are useful for flight simulators, because pilots are trained continuously on a single aircraft. Path planners must be broader in functionality than simulators, because they must be able to model and plan paths for multiple aircraft. Therefore the calculation must be universal to all aircraft.

This research focused on finding calculations that would provide enough accuracy to model the UAV flight characteristics while being computationally efficient. The calculations must be broad enough to encompass all different types of aircraft with minimal configuration of the model. Load factor (G-force) and stall velocity were chosen as the aircraft characteristics to optimize because all aircraft have these constraints. Path planning for new aircraft would require specifying characteristics attainable from manufacturing specifications, such as load factor, stall velocity, and maximum velocity. Computational efficiency is an important consideration when designing an online path planner, because it must be close to real time. First order

flight mechanics equations (velocity) are powerful enough to give accurate results without being overly complicated. Second order calculations (acceleration) provide more accurate models but are more complicated and require extra computational power. Based on the current level of flight characteristics seen in path planners, it was determined that second order equations would not add significantly to the results. Clancy provides a series of equations for flight mechanics that meet the desired requirements and were implemented in this path planner [55]. The basic equations are:

1. Pull-outs
2. Climbing
3. Gliding
4. Steady, Level, Co-ordinate Turn

Pull-outs

Pull-outs occur when an aircraft is diving along a vertical path and must maneuver to recover from the dive and regain a level or climbing flight. The flight path of a pull-out will be a vertical curve but not always circular. However, if the flight path is assumed to be circular, the maximum possible force experienced by the aircraft along any point of the dive can be calculated instead. The same equations can be used if the aircraft is flying erect, i.e. not upside down, where the aircraft will experience negative forces.

Figure 25 shows the free body diagram depicting a vertical trajectory pulling out of a dive. W is the weight of the aircraft, V is the velocity, L is the lift force on the aircraft, R is the radius of the path, and θ is the angle from vertical. An aircraft flying erect would look similar with everything flipped across the horizontal axis, with Lift still pointing up and weight still pointing down. To calculate the load factor (n) experienced by the aircraft, based on Clancy's Equations 26 and 27, the following variables must be known: velocity (V), weight (W), radius of path (R), gravitational constant (g), and the angle from vertical (θ). Theta (θ) will be between 0 and 90 degrees, with 0 being the bottom of the path with the aircraft flying horizontally and 90 being an aircraft in a completely vertical dive.

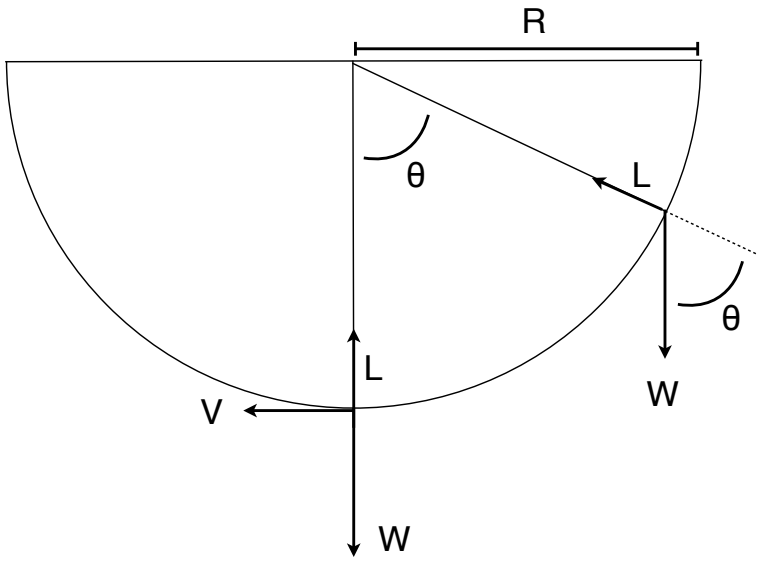

Figure 25: Pull-out condition free body diagram

$$L = \frac{V^2}{(Rg + \cos\theta)}W \quad (26)$$

$$n = \frac{L}{W} \quad (27)$$

Climbing

Climbing is the act of an aircraft climbing vertically into the sky. This is a flight maneuver common across all aircraft. Clancy's equations deal with a straight path with a constant inclined angle. The forces acting on the aircraft, seen in Figure 26, consist of Lift (L) normal to the flight path, thrust (T) and drag (D) parallel to the path, and weight (W) acting in a vertical downward direction. The aircraft is climbing at an angle of γ. The aircraft velocity (V) is along the flight path with the climbing velocity (Vc) in the vertical direction.

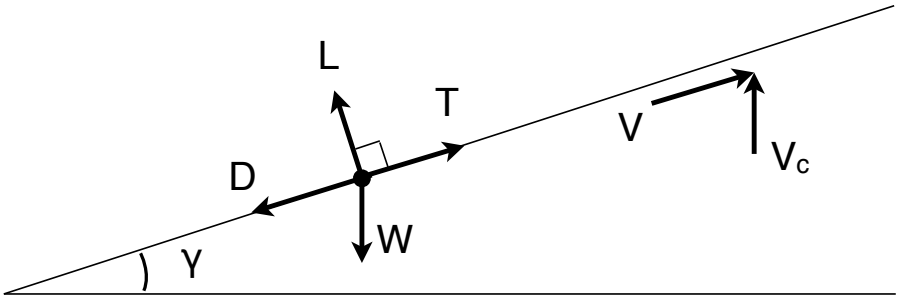

Figure 26: Climbing condition free body diagram

Using Equation 28 the resulting forces experienced by the aircraft as well as the climbing velocity can be calculated. Lift is another flight mechanics characteristic that is crucial when generating alternate paths in the path planner. Lift (L) is calculated by using the climbing angle (γ) and the weight of the aircraft, considering our system is in equilibrium the forces must balance. The Lift is then converted into a constraint useful to the path planner, load factor (n) calculated using Equation 27.

$$L = W \cos\gamma \quad (28)$$

$$n = \frac{L}{W} \quad (27)$$

Gliding

Gliding is similar to climbing, but done in the opposite direction. Steady glide is a descent without engine power along a straight path. Figure 27 shows the free body diagram of the glide maneuver. The glide angle (γ) is the angle of the flight path relative to horizontal. The aircraft experiences slightly different forces than climbing, because there is no thrust component. The aircraft experiences Lift (L) perpendicular to the flight path, Drag (D) parallel to the flight path, and Weight (W) vertically downward. The sinking speed (V_s) is the vertical component of speed defining how quickly the aircraft is moving in the vertical direction expressed through Equation 29.

Equation 28 shows how to relate the Weight (W) of the aircraft to the lift force (L). Similar to the climbing relationship, the weight and lift are related by the glide angle (γ). Since the aircraft is in steady state, the weight and lift forces must balance. The lift must then be

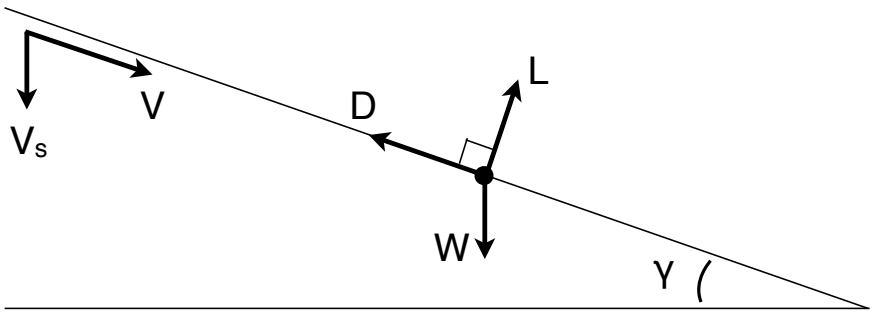

Figure 27: Gliding condition free body diagram

converted to the load factor (n) using Equation 27, to use as a path planner constraint.

$$L = W \cos \gamma \tag{28}$$

$$n = \frac{L}{W} \tag{27}$$

$$V_s = V \sin \gamma \tag{29}$$

Steady, Level, Co-ordinate Turn

The steady, level, co-ordinate turn is an idealized condition for aircraft turning where the aircraft experiences no sideslipping. A co-ordinate turn is by definition a turn without sideslip. This type of turn is a good approximation for aircraft turning and is aerodynamically the most efficient way to make a level turn. Path planners that use a minimum turning radius in their calculations, set a constant value the aircraft cannot exceed. However this is a simplification of a turning condition, where the minimum turning radius changes with the speed of the aircraft. The stall speed of an aircraft also changes when turning requiring the minimum flight speed to be altered for a safe flight path.

Figure 28 shows the free body diagram for the steady, level, co-ordinate turn condition. The circle in the center is the aircraft with the black lines at an angle representing the wings. The weight (W) is in the vertical downward direction, lift force (L) is still normal to the flight path, which is now rotated by the angle of bank (ϕ) from vertical. The angle of bank denotes the rotation of the aircraft around the roll axis from horizontal.

Equation 30 represents the relationship between the weight (W) of the aircraft and the lift force (L) experienced by the aircraft in a steady, level, co-ordinate turn. The unknown in the relationship is the angle of bank of the aircraft (ϕ). Equation 31 relates the lift (L), the bank angle (ϕ), with the velocity (V) and the turn radius (R). To calculate the lift experienced by the aircraft, both Equations 30 and 31 must be solved simultaneously, with the result shown in Equation 32. Once the lift has been calculated, the load factor (n) can be calculated using Equation 27. With the known stall velocity for the aircraft and the newly calculated load factor, Equation 33 can be used to find the effective stall velocity for the current turn.

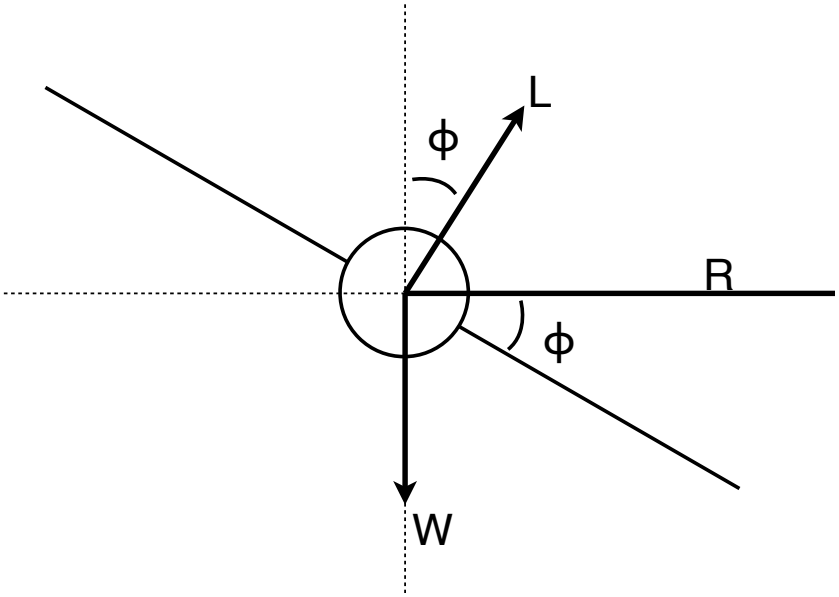

Figure 28: Steady, level, co-ordinate turn free body diagram

$$W = L\cos\varphi \tag{30}$$

$$L\sin\varphi = \frac{V^2}{Rg}W \tag{31}$$

Solved Simultaneously:

$$L = W \frac{1}{\cos\left(\tan^{-1}\frac{V^2}{Rg}\right)} \quad (32)$$

$$n = \frac{L}{W} \quad (27)$$

$$V_{stall} = V_s\sqrt{n} \quad (33)$$

The four flight mechanics equations described above require a few bits of information to solve. The aircraft's position, velocity, and turn radius are required at every point along the flight path. The weight of the aircraft is also required for most calculations. Once the four flight mechanics equations are calculated, the path planner needs to determine if the path is feasible. The maximum load factor and stall velocity are two constraints consistent with all airplanes and will be used to determine if the paths are feasible.

The position and velocity of the aircraft were calculated using the equations describe in the section "B-splines." The turn radius was calculated using the position on the curve, the next point on the curve, and the first derivative. Figure 29 shows how two positions and a derivative can be used to calculate the turning radius. The flight path of the aircraft is represented by the curve at the top of Figure 29. The current aircraft position is represented by the dot on the left side of the path and the next location on the path is represented by the dot on the right side. Vector V represents the derivative of the aircraft's position. A cord is drawn between the beginning and

ending point on the curve. The cord is bisected giving lengths of l. The angle ϕ can be determined by calculating the difference between the horizontal cord and the derivative vector V. Drawing a line perpendicular to the vector V and intersecting with a perpendicular line located a length of l from the first point on the curve, represents the center of a circle. From here, trigonometry can be used to calculate the hypotenuse of the triangle, which is the radius of the curve R.

Changes to Optimization Formulation

Evaluating the flight mechanics of the flight paths requires modifying the original optimization formulation. The original formula evaluated fuel efficiency, reconnaissance, and safety. By changing the constants, K_1, K_2, K_3, it was possible to return paths focusing on a certain characteristic. For example, if K_1 is set to 0.9, K_2 is set to 0.05, and K_3 is set to 0.05, the returned path will be more fuel-efficient. The terrain cost was utilized in the problem formulation differently. Instead of using a constant to balance terrain violation with the other constraints, the terrain violation added a large cost to the overall cost value. Using this technique, violations of the terrain were essentially eliminated. Flight characteristics should operate in a similar fashion because violating the maximum load factor could result in structural failure to the vehicle. Equation 34 shows the modification to the original optimization formulation. The flight mechanics cost (F) was added to the end of the equation to provide a cost every time the flight mechanics equations were violated.

$$C = K_1 C_F + K_2 C_R + K_3 C_T + T + F \tag{34}$$

The value of F was calculated by evaluating the characteristics of each individual flight path for violation in altitude, velocity, turning radius, and load factor. All four flight mechanics conditions were evaluated, pull out, glide, climb, and level turning, for every point along the flight path. Once the load factor was determined for the path, Equation 35 was used to create the load factor cost. The load factor was evaluated for every point along the path with $LoadFactor_i$ representing the load factor for a single point on the flight path and $numC$ representing the total number of points on the flight path.

$$LoadFactorCost = \sum_{i=1}^{numC} \begin{cases} 0 & , LoadFactor_i < MaxLoadFactor \\ \left| \frac{LoadFactor_i}{MaxLoadFactor} \right| & , LoadFactor \geq MaxLoadFactor \end{cases} \quad (35)$$

The stall velocity is another important flight characteristic that must be considered when creating feasible paths. All aircraft have a stall velocity associated with their design and payload. When an aircraft turns, the effective stall velocity goes up. This requires an aircraft to travel at a higher rate of speed to avoid stalling. Every point along the flight path was evaluated to determine if the instantaneous velocity dropped below this effective stall velocity. If the instantaneous velocity ($velocity_i$) drops below the effective stall velocity ($effectiveStall_i$), a cost of 5.0 is added. Equation 36 shows the summation of all stall velocity constraints into a single cost function ($StallVelocityCost$). Each point on the path must be evaluated and added together to get the total cost remembering that the $effectiveStall_i$ changes for each

point on the path depending on the turn radius. Equation 37 was similarly formatted and represents the cost for violating the maximum altitude ($maxAlt$) capable of being flown by the specified aircraft. The altitude of the aircraft was compared at every point along the flight path ($altitude_i$) and summed to determine the total cost.

$$StallVelocityCost = \sum_{i=1}^{numC} \begin{cases} 5.0, & velocity_i \leq effectiveStall_i \\ 0.0, & velocity_i > effectiveStall_i \end{cases} \quad (36)$$

$$MaxAltitudeCost = \sum_{i=1}^{numC} \begin{cases} 0.0, & altitude_i \leq maxAlt \\ 5.0, & altitude_i > maxAlt \end{cases} \quad (37)$$

All three flight characteristic costs are then summed to determine the total flight mechanics cost for the flight path (F). The magnitude of F should be between 0 and 100 to match magnitudes with the other cost in the optimization problem formulation. Equation 38 shows the addition of all flight components to arrive at the total cost for the flight path.

$$F = LoadFactorCost + StallVelocityCost + AMaxltitudeCost \quad (38)$$

Visualization of Vehicle Dynamics in 3D

With an acceptable path calculated, the path must now be visualized to the operator. The “Path Visualization” section discusses all the different methods explored for displaying paths. Since Battlespace is designed for interaction in an immersive system, a 3D method of visualization is needed.

Originally Battlespace's visualization of the UAV flight paths was waypoint centric. All flight path information is displayed to the operator in terms of waypoints. Figure 8 and 13 shows the original visualization of the flight path as a yellow fence. The fence is comprised of alternating yellow and transparent horizontal bars. Each horizontal bar is used to display the height of the UAV as well as the direction, with each bar representing 1,000ft of altitude. The flight path itself is visualized by drawing the fence directly between two waypoints.

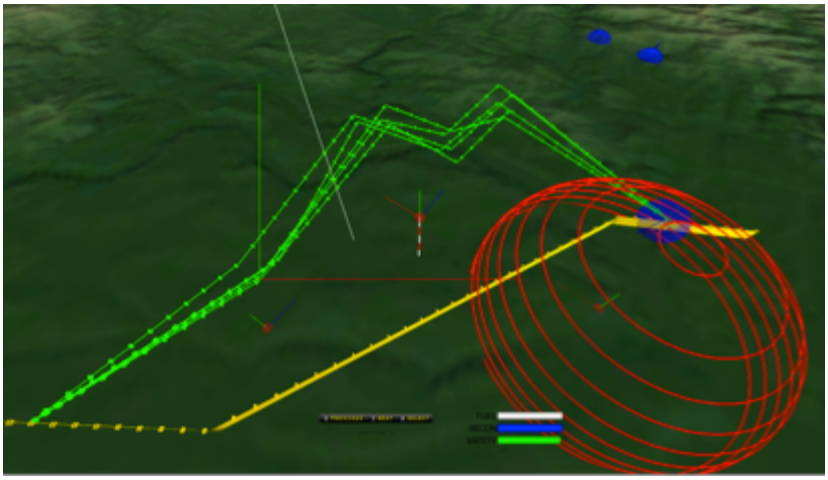

Figure 29: Original visualization of alternate paths

The alternate paths returned by PSO Path Planner are also visualized with the green path lines in Figure 29. The waypoints can be found by looking for sharp turns in the path itself. Alternate paths are visualized the same way as the actual (yellow) path, but drawing a straight line between the two waypoints.

Representing UAV flight paths by connecting straight lines between the waypoints is an efficient method of visualizing the paths. While this technique is efficient, it does not accurately depict the UAV path through those waypoints. Most path planers create a set of waypoints for the UAV to follow with the assumption that the aircraft

follows the waypoints as close as possible. Figure 30 shows a 2D representation of the original flight path visualization and the new b-spline interpolation flight path. The pink dots represent the waypoints given to a UAV to follow. The red line represents the original flight path visualization techniques where a straight path is drawn between two adjacent waypoints. The green line represents a NURBS path interpolated through the given waypoints using universal interpolation as described in the “B-splines” section.

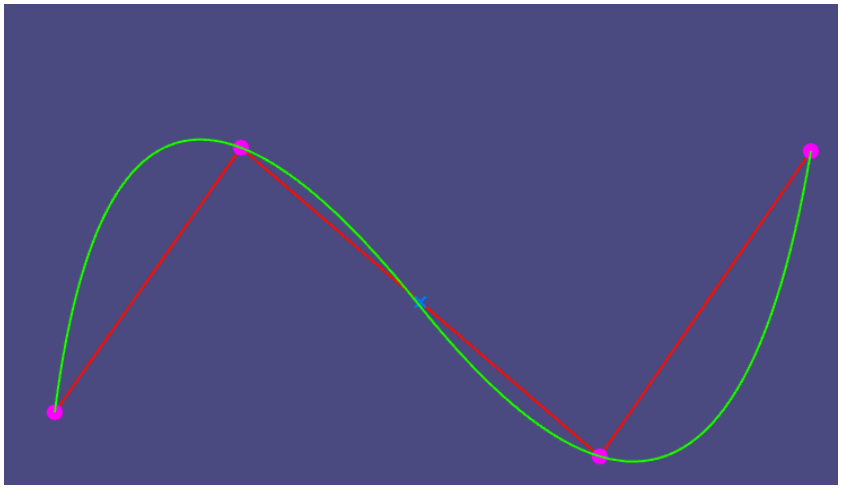

Figure 30: 2D representation of difference between original visualization and B-spline interpolation

The Figure 30 illustrates how much the NURBS flight path can vary compared to the traditional linear flight path visualization. NURBS representations can vary depending on the degree of the NURBS curve. The higher the degree, the less impact an individual waypoint would have on the flight path. The higher degree would translate into a flight path with fewer turns. A lower degree curve provides allows each waypoint to have more control of the path, allowing sharper turns in the path.

NURBS allow a more accurate representation of the UAV flight path by eliminating the large differences in flight path location visualized in Figure 30. The NURBS representation more accurately depicts the looping motion taken by a UAV to travel the given waypoints. Visualizing the path using this technique helps cut down on the amount of the path an operator must construct in their mind. This also prevents the operator from accidentally selecting a path traveling through a threat when the linear representation shows the flight path avoiding the threat.

Chapter 4 – Results and Evaluation

Changes to the path generation method, cost function improvements, and visualization changes were evaluated to determine the potential benefits to UAV path planning and operator interaction. Both quantitative and qualitative methods were used to determine if the research goals were achieved. The research goals were improving the cost function by incorporating flight mechanics to provide feasible paths and developing a visualization strategy to accurately display the flight path. Both of these goals need to be achieved at close to real time to maintain the online path planner functionality.

To evaluate the impact of flight mechanics, five scenarios were created, each involving varying numbers and types of threats. Scenarios are a set of waypoints for the UAV to travel with the path traveling near or through areas with threats such as SAM sites or enemy fighters. The UAV traveled different routes (waypoints) and encountered different types of threats (ground or air) depending on the scenario.

Each scenario was run four times for the original path planner and four times for the new path planner with flight mechanics. The numerical results for the individual cost components and the total cost were recorded and compared to determine the impact. The paths were visually inspected to qualitatively determine the benefits of visualizing the curved path versus traditional waypoints. Paths were considered feasible if all sharp turns in the path were eliminated. Infeasible or sharp turns would be a set of waypoints creating an acute angle. Ideally, paths containing sharp turns

would be completely eliminated, however if PSO Path Planner does not find any feasible paths based on the flight mechanics cost, paths may be returned with infeasible or sharp turns.

Scenarios

The scenarios are chosen to represent a variety of possible situations UAVs could face in the act of duty. The scenarios contain an increasing level of complexity starting with a single ground threat, moving up to the most complex scenario with two ground threats and one air threat. The scenarios are summarized in Table 1.

Table 1: List of evaluation scenarios

Scenario	Description
1	One Ground Threat
2	One Ground Threat
3	One Aerial Threat
4	One Ground and One Aerial Threat
5	Two Ground and One Aerial Threat

Scenarios 1 and 2 are very similar and can be visualized by Figure 31 showing the flight path of the UAV, represented by the yellow fence, traveling through the threat zone of a SAM site. There is also a waypoint positioned inside the threat zone, which the UAV will attempt to perform reconnaissance on. The differences between Scenario 1 and 2 are the position of the ground threat and the initial path taken by the UAV. Scenario 3 is similar to Scenarios 1 and 2, except the threat is airborne instead of ground based. An airborne threat provides the UAV path planner more

options to avoid a threat because the UAV can go under the threat. Figure 32 shows Scenario 3 with both the UAV (bottom of image) and the airborne threat (top of image). The path planner must account for performing reconnaissance on the waypoints in the vicinity of the threat.

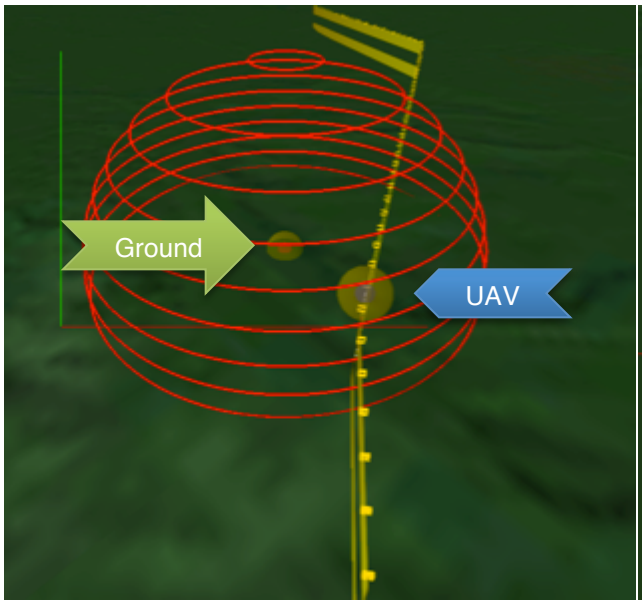

Figure 31: Scenario 2, with one ground threat

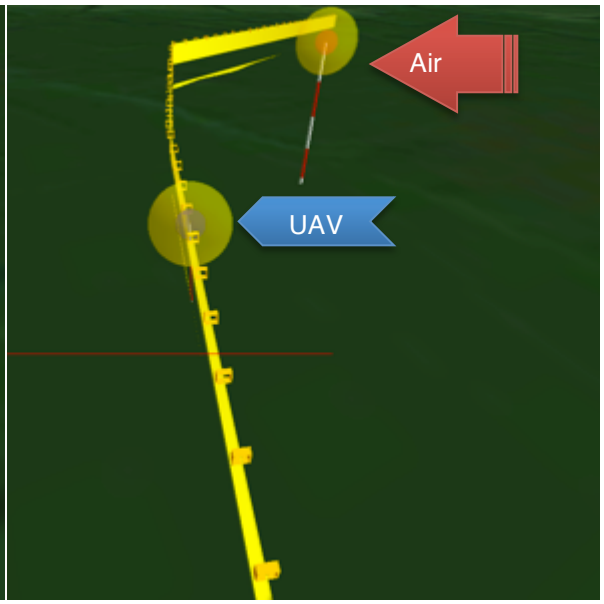

Figure 32: Scenario 3, with one aerial threat

Scenarios 4 and 5 combine the potential for both ground and aerial threats. These two scenarios are designed to be more complex and challenge the path planner. Scenario 4 is visualized in Figure 33 with one ground threat and one aerial threat. The path planner must take into consideration the reconnaissance needing to be performed on the waypoints along the initial path. The ground threat will be identified first, but the aerial threat will also need to be considered because it is within range of the original path. Scenario 5, depicted in Figure 34, has two ground threats and one

aerial threat with their own distinct threat zones. Reconnaissance is needed on two waypoints along the initial path.

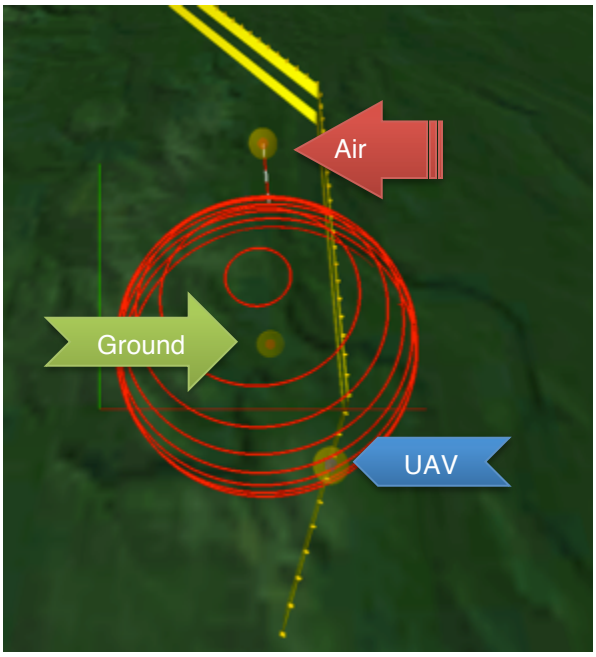

Figure 33: Scenario 4, with one ground and one aerial threat

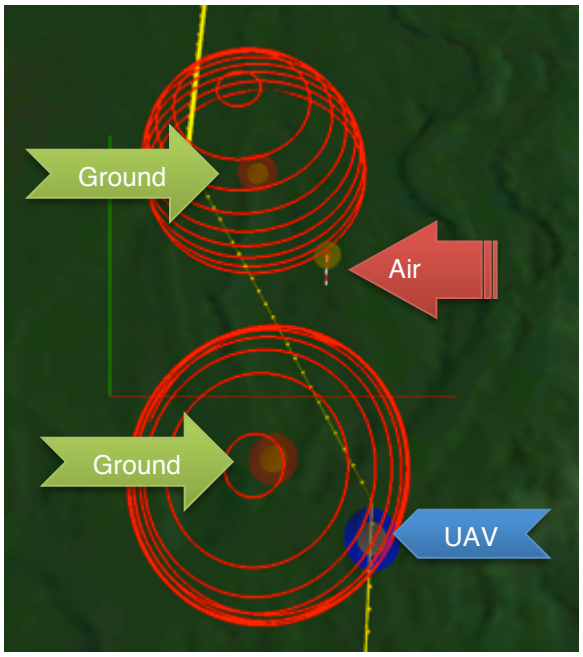

Figure 34: Scenario 5, with two ground threats and one aerial threat

Each scenario presents different challenges to the path planner depending on the type of threats and the number of threats. Air threats limit the altitude for alternate paths compared to the ground threats, but also provide the ability to travel underneath the threat. The more threats in a scenario the more maneuvering the aircraft must do to avoid all the threats and consequently the greater the possibility for violating flight constraints.

UAV Flight Specifications

To evaluate the flight mechanics component of the path planner, a suitable aircraft must be defined. The flexibility provided by the flight mechanics equations allows

them to be used for all types of aircraft. The weight, maximum load factor, stall velocity, and maximum altitude are required for this application. While most of this information is readily available to the public, the maximum load factor is not, so an approximation is required for this evaluation. An unmanned vehicle's maximum load factor is determined by the structural limits of the vehicle and not the limitations of the pilot. The structural limitations of UAVs could be considered proprietary information and could explain why the information was not publically available. Information was found for two different aircraft, the Boeing X-45 and the General Atomics MQ-1 Predator. Due to information not being readily available for a single UAV the information found for these two vehicles was combined to provided an approximate model of a UAV.

The fully loaded weight of a Boeing X-45 is 5528 kilograms (kg) [63]. The stall velocity for the MQ-1 Predator is 100 kilometers per hour (km/hr) [64]. The maximum altitude for the X-45 is 10,670 meters (m) [63]. An approximation of 6 meters per second squared (m/s^2) or 6 standard gravities (g) was used for the maximum load factor based on approximation of the maximum load factor for fighter pilots (~10g) and the types of long range missions UAVs typically fly. The UAV specifications used in the evaluation are summarized in Table 2.

Table 2: UAV aircraft specifications for evaluation

Value	Specification
5528 kg	Weight
100 km/hr	Stall Velocity
10,670 m	Maximum Altitude
6.0 g	Maximum Load Factor (g-force)

Testing Parameters

PSO Path Planner runs a separate optimization problem to generate each group of five paths. By changing the component weights in Equation 10 and 34 (K_1, K_2, K_3) each separate optimization run can focus on fuel efficiency, threat avoidance, or reconnaissance. Table 3 gives the breakdowns of the component weights and how they were balanced for each path type.

Table 3: Component weights

Path Type	Fuel Weight, K_1	Recon Weight, K_2	Threat Weight, K_3
Fuel Efficient	0.90	0.05	0.05
Reconnaissance	0.05	0.90	0.05
Threat Avoidance	0.05	0.05	0.90

To establish a baseline for comparison, all five scenarios were run first using the original problem formulation (Equation 10) without the added flight mechanics component. PSO path planning was performed four times for each scenario with each run producing fifteen alternate paths.

To evaluate the results of adding flight mechanics to the path planner, all five scenarios were run using the modified problem formulation (Equation 34) that includes the flight mechanics component. PSO path planning was performed four times for each scenario, generating 15 unique paths on each run. The difference between these paths and the baseline paths are the added flight mechanics component, which applies a cost any time the flight path exceeds the load factor, altitude, or velocity constraints. This component acts similar to the terrain component, where the flight paths may violate the constraint, but a large cost reduces the probability of a flight constraint violation.

The fifteen paths for each run were broken into three path types (fuel efficient, reconnaissance, safety) as seen in Table 3 with 5 paths in each of the three groups. Each of the fifteen paths were broken down into the individual components (fuel, threat, recon, terrain, flight mechanics) and evaluated. Breaking the total cost up into the individual components allows for better comparison of the interactions between characteristics of the paths. Determining how flight mechanics affects the individual components is important to understanding the gains from this method. The total cost provides overall gains but does not explain where the true benefits are.

PSO Path Planner Results

The paths are evaluated quantitatively by comparing the component costs of each path generated between the original PSO Path Planner and the new PSO Path Planner with flight mechanics. The total costs of the paths are expected to increase

slightly due to adding another constraint, the flight mechanics. The paths are evaluated qualitatively by inspecting the resulting paths and determining whether the paths look feasible and accurate. Visually, the paths should be smoother with no sharp turns or sharp dives to avoid an obstacle.

For example, Scenario 2 was first run using the original PSO Path Planner optimization formulation, Equation 10. This scenario was composed of a single ground threat along the original path of the UAV. Figure 35 shows a side view of Scenario 2 and Figure 36, shows a top down view of the same scenario. The original path planner was run, with the top five results being recorded for each of the three conditions (fuel efficient, threat avoidance, and reconnaissance). The paths were visually inspected to determine the feasibility of the paths. After recording the results, the new version of PSO Path Planner, with the flight mechanics equations, was run for comparison.

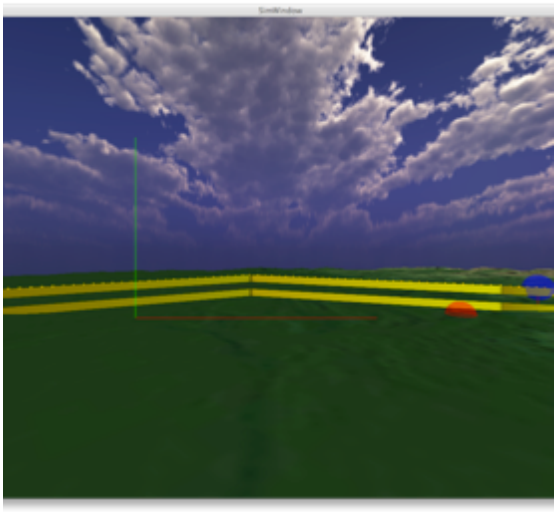

Figure 35: Scenario 2 side view

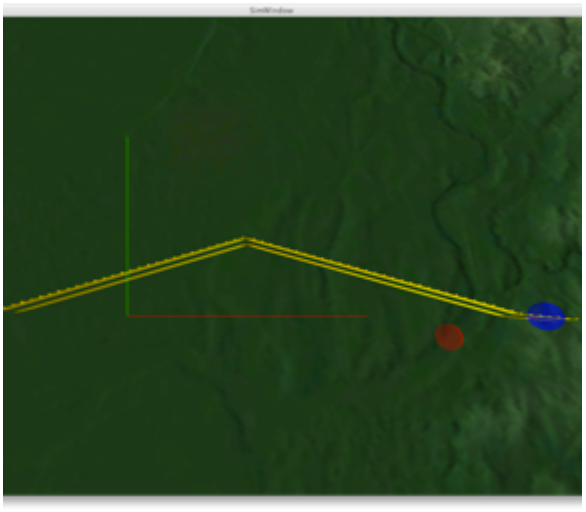

Figure 36: Scenario 2 top view

Figure 37 visually shows the results of the original Path Planner formulation without the flight mechanics component. The paths produced accomplish the goal of avoiding the threats by planning paths outside or near the outside of the threat zones. The reconnaissance and fuel objectives are also met, as seen by the paths staying close to the original path. At no time does the path travel through the terrain causing a terrain cost. However, the paths are being generated with sharp turns and in one case, the flight path makes a series of sharp turns and cross itself. These paths could be considered unfeasible due to their sharp turns.

Figure 38 shows the results of the new PSO Path Planner with flight mechanics. Visually the paths appear similar to the original problem formulation with the path location around the side of the threat. The reconnaissance and fuel goals are both met by planning paths close to the original path, nor is the terrain violated. The

difference is noticeable in the lack of sharp turns. All the paths shown have obtuse angles, which indicate more gradual turns.

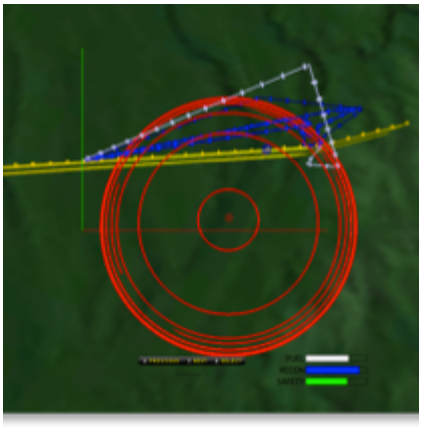

Figure 37: Original PSO Path Planner results

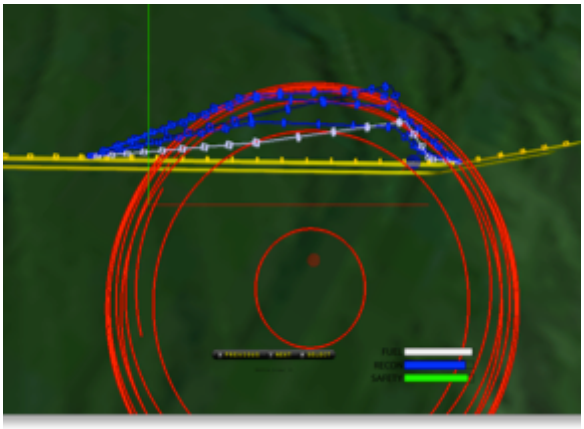

Figure 38: PSO Path Planner with flight mechanics

Trends similar to the results described above are seen in all five scenarios. Both the original PSO Path Planner and the new PSO Path Planner with flight mechanics achieve their goals. The three main constraints (fuel, recon, safety) were met while satisfying the new flight mechanics constraint. The major difference between the two types of PSO Path Planner is the decrease in the sharpness of the turns in all three dimensions.

The components composing the total cost of each path were also recorded to quantitatively compare the results of both versions of the PSO Path Planner. Through comparison of these components, the impact of the flight mechanics on the

types of paths returned can be determined. The component costs break down for all 15 paths (5 fuel, 5 reconnaissance, 5 safety) for the original PSO Path Planner are shown in Table 4 and the results from the new PSO Path Planner with flight mechanics are in Table 5. Remember that smaller numbers represent a smaller cost and are therefore better. Both the original and the new path planner were run four times for each scenario. Table 4 and 5 were selected as a representative example of the results. The original PSO Path Planner calculated the flight mechanics cost component for each path for comparison purposes. The flight mechanics cost component was not applied to the total cost nor was it a factor in solving the optimization problem.

Table 4: Breakdown of component costs for Scenario 2 Run 1 without flight mechanics

Path	1	2	3	4	5
Run 1	Fuel = 90%				
Total Cost	11.122	11.169	11.221	11.251	11.255
Fuel	10.132	10.052	10.081	10.112	10.044
Recon	24.010	25.024	27.023	26.083	28.968
Safety	16.050	17.430	15.945	16.922	15.352
Terrain	0.000	0.000	0.000	0.000	0.000
Flight Mechanics	29.005	21.000	26.006	22.036	23.000
Path	6	7	8	9	10
	Recon = 90%				
Total Cost	25.771	28.473	28.610	28.661	28.756
Fuel	12.064	9.978	10.512	10.437	10.728
Recon	26.855	30.187	30.204	30.223	30.428
Safety	19.965	16.125	18.020	18.784	16.685
Terrain	0.000	0.000	0.000	0.000	0.000
Flight Mechanics	23.393	45.006	44.408	47.177	35.747
Path	11	12	13	14	15
	Safety = 90%				
Total Cost	11.770	11.792	11.922	12.011	12.170
Fuel	10.007	10.101	10.166	10.251	10.065
Recon	45.387	45.742	48.263	49.976	45.132
Safety	10.000	10.000	10.000	10.000	10.456
Terrain	0.000	0.000	0.000	0.000	0.000
Flight Mechanics	33.004	49.070	23.012	27.045	47.076

Table 5: Breakdown of component costs for Scenario 2 Run 5 with flight mechanics

Path	1	2	3	4	5
Run 5	Fuel = 90%				
Total Cost	11.310	11.395	11.455	11.481	11.486
Fuel	10.431	10.380	10.296	10.613	10.560
Recon	25.921	28.165	31.719	26.605	27.017
Safety	12.512	12.906	12.055	11.989	12.615
Terrain	0.000	0.000	0.000	0.000	0.000
Flight Mechanics	0.000	0.000	0.000	0.000	0.000
Path	6	7	8	9	10
	Recon = 90%				
Total Cost	33.523	34.333	39.595	41.289	41.396
Fuel	14.271	13.158	15.533	13.479	14.254
Recon	26.217	36.796	33.658	31.794	44.481
Safety	12.397	11.174	12.909	11.566	13.017
Terrain	0.000	0.000	0.000	0.000	0.000
Flight Mechanics	8.594	0.000	7.881	11.423	0.000
Path	11	12	13	14	15
	Safety = 90%				
Total Cost	11.868	11.924	12.044	12.066	12.095
Fuel	11.419	10.858	11.348	11.825	11.298
Recon	45.932	47.615	49.527	49.502	50.595
Safety	10.000	10.000	10.000	10.000	10.000
Terrain	0.000	0.000	0.000	0.000	0.000
Flight Mechanics	0.000	0.000	0.000	0.000	0.000

Table 4 and 5 represent two individual runs, one for the original path planner and one for the new path planner. The data collected in the table represents the total cost and breakdowns of the total cost component being evaluated by PSO.

The “Total Cost” is the overall cost of the path chosen, including the fuel, reconnaissance, safety, and terrain cost. In the case of the fuel efficient paths, the fuel cost component makes up 90% of the “Total Cost”. The other two components,

reconnaissance and safety, equally comprised the other 10%. The values in the table are the cost components before any scaling, multiplying fuel by 0.9, and summing to the total cost component. The “Flight Mechanics” rows indicate the evaluated flight mechanics cost associated with the given paths.

The terrain components of both the original and the new path planner are zero for all returned paths. This confirms what was inspected visually, that all paths were feasible and above the terrain. The “Total Costs” between the original paths and the new paths are different, but there is a noticeable increase across the board for these two runs when comparing the same types of paths together (e.g. original fuel efficient paths with new fuel efficient paths). The largest change is in the “Flight Mechanics” cost component, where the original path planner returned paths in the range of 20-50. This cost is calculated using the flight mechanics equations but is not applied to the total cost. This value indicates the level of which the returned paths are infeasible in terms of the flight constraints. The new path planner with flight mechanics included in the optimization problem formulation returns paths with “Flight Mechanics” cost in the range of 0-10. This indicates the original path planner was returning infeasible paths to the operator in the case of these two runs.

The original PSO Path Planner was ran four times with the total cost being averaged for the three dominant variables, shown in Table 6. The new PSO Path Planner with flight mechanics was also ran four times averaging the total cost for the three dominant variables. The average total cost for both the original and new path

planner was compared to determine the percentage increase in total cost when considering flight mechanics. Remember a smaller total cost is better.

Table 6: Average Total Cost values for Scenario 2

Dominant Variable	Total Cost w/o Flight Mechanics	Total Cost w/ Flight Mechanics	% Increase
Fuel	11.033	12.118	9.83%
Recon	29.711	42.856	44.24%
Safety	11.960	14.127	18.11%

The data for Scenario 2 shows an increase in the total cost of the path when flight mechanics are included in the optimization problem formulation. There is a smaller increase in the total cost with paths focused on fuel efficiency. The largest increase at 40% was observed for paths weighted for reconnaissance. This indicates that there is a greater impact of flight mechanics on reconnaissance.

The increase to the reconnaissance cost comes from the flight mechanics restrictions on sharp turns. The reconnaissance cost is lowest when the flight path travels through all waypoints. Limiting the UAVs ability to make sharp, often infeasible, turns limits the UAV's ability to travel within close proximity to the waypoints. Consequently the reconnaissance cost increases. The other two components, fuel and safety, are not affected by the restrictions imposed by flight mechanics on sharp turns because those components do not benefit from this ability.

Table 7 presents a breakdown of all the components of the total cost, which determines where the majority of the costs were accrued. The first column of Table 7

breaks down the top fifteen paths by the dominant variable – which variable makes up 90% of the total cost function. Reminder, smaller numbers are better.

Table 7: Average totals for eight runs of Scenario 2 for three different types of paths

	Without Flight Mech			With Flight Mech				% Increase		
Dominant Variable	**Avg Fuel**	**Avg Recon**	**Avg Safety**	**Avg Fuel**	**Avg Recon**	**Avg Safety**	**Avg FM**	**Avg Fuel**	**Avg Recon**	**Avg Safety**
Fuel	9.99	25.82	15.09	10.60	32.02	12.49	0.00	6.20%	24.01%	-17.29%
Recon	13.49	31.32	16.95	16.95	38.06	16.01	6.95	25.64%	21.52%	-5.53%
Safety	11.24	46.76	10.07	13.56	56.31	10.56	1.13	20.71%	20.41%	4.89%

It is important to focus on the cells that match the dominant variables with the corresponding components, i.e. a dominant fuel variable with the average fuel component for the paths returned. These are the dominant values making up 90% of the total cost. Increases of 6.20%, 21.52%, and 4.89% are witnessed in the three dominant variables. Increases in these three key areas indicate incorporating flight mechanics in the cost function increases the total cost compared to the original path planner.

Table 7 shows a decrease in the safety component cost for the other two dominant variables (fuel and reconnaissance). This indicates the safety of the path improves when flight mechanics is included in the problem formulation. An improvement in safety is a desirable solution, however the cause of this improvement is not directly attributed to the flight mechanics. The focus component, either fuel or reconnaissance, has the most weight and is the component being optimized. The component with the most weight has the ability to overpower the other components

in an attempt to optimize the focus component's constraints. Even with a decrease in the safety cost, the total cost increases for these paths. It can therefore be concluded that including flight mechanics will not guarantee an increase in safety.

Five Scenarios with Individual Components

Determining the effects of flight mechanics in PSO Path Planner requires a comparison of multiple scenarios. Scenarios 1-5 were chosen to challenge PSO Path Planner with varying levels of difficulty through combinations of ground and air threats. Evaluating the results for all five scenarios allows conclusions to be made for the impact of flight mechanics on the paths generated by PSO Path Planner. Comparing the changes in individual components allows conclusions to be drawn about how flight mechanics affects the path planner. Specifically, how does it affect fuel, reconnaissance, and safety costs? Comparing the total costs of paths from Scenarios 1-5 provides compelling evidence that flight mechanics enhances the paths generated by ensuring feasibility.

The comparison in Table 8 reveals a significant increase in the cost component values of the returned paths when flight mechanics are considered by as much as 20%. Remember that smaller numbers are better. With flight mechanics considered almost all paths returned exhibit no cost or a negligible cost for flight mechanics indicating the paths returned satisfy the load factor, velocity, and altitude constraints defined. Given that the flight mechanics paths return negligible cost for flight mechanics and an increase in the individual cost components (fuel, recon, safety), it

can be concluded that the original paths returned were not feasible with regards to flight constraints.

Table 8: Average total cost of all five scenarios considering three types of paths, 8 runs each

Scenario1	Without FM			With FM				% Increase		
Dominant Variable	**Avg Fuel**	**Avg Recon**	**Avg Safety**	**Avg Fuel**	**Avg Recon**	**Avg Safety**	**Avg FM**	**Avg Fuel**	**Avg Recon**	**Avg Safety**
Fuel	10.23	27.39	21.58	10.81	28.75	17.21	0.05	5.63%	4.94%	-20.25%
Recon	12.64	24.48	22.18	14.84	26.17	14.39	0.00	17.44%	6.88%	-35.11%
Safety	12.04	29.43	10.09	12.76	33.76	10.09	0.00	6.04%	14.73%	-0.06%
Scenario2										
Fuel	9.99	25.82	15.09	10.60	32.02	12.49	0.00	6.20%	24.01%	-17.29%
Recon	13.49	31.32	16.95	16.95	38.06	16.01	6.95	25.64%	21.52%	-5.53%
Safety	11.24	46.76	10.07	13.56	56.31	10.56	1.13	20.71%	20.41%	4.89%
Scenario3										
Fuel	10.09	33.55	30.08	10.19	28.19	30.26	0.00	1.04%	-15.96%	0.58%
Recon	12.98	30.38	34.67	11.55	30.38	31.73	0.00	-10.97%	0.00%	-8.47%
Safety	13.96	95.00	13.19	14.52	104.16	14.98	0.22	3.98%	9.64%	13.55%
Scenario4										
Fuel	10.56	32.38	19.83	10.81	36.84	17.82	0.00	2.35%	13.79%	-10.16%
Recon	11.09	34.51	19.29	11.09	38.86	17.43	0.00	0.03%	12.63%	-9.62%
Safety	11.64	53.84	10.00	11.65	54.78	10.32	0.00	0.07%	1.75%	3.20%
Scenario5										
Fuel	10.66	37.67	19.20	10.57	38.45	18.74	0.00	-0.89%	2.08%	-2.42%
Recon	10.38	32.85	20.19	11.06	40.10	17.60	0.00	6.53%	22.07%	-12.84%
Safety	11.40	58.79	10.00	11.23	54.93	10.00	0.00	-1.49%	-6.57%	0.00%
Total Avg										
Fuel	10.30	31.36	21.16	10.60	32.85	19.30	0.01	2.87%	5.77%	-9.91%
Recon	12.12	30.71	22.65	13.10	34.71	19.43	1.39	7.73%	12.62%	-14.32%
Safety	12.05	56.76	10.67	12.74	60.79	11.19	0.27	5.86%	7.99%	4.32%

Once again there are anomalies in the table showing a decrease in the cost component specifically in regards to safety, indicating the paths returned were safer when including flight mechanics in the problem formulation. Most of these occur when safety is not the dominant variable for the path. These results are misleading

because the component is such a small component of the total cost. The reason for this is inherent in the optimization method itself. During the optimization process, the path planner will tend to optimize the dominant variable while still giving some preference to the other non-dominant variables. In this case, when either fuel efficiency or reconnaissance is the dominant variable, safety will not be of high priority, thus generating paths that are inside the threat zone. However, in optimizing the paths for these dominant variables, the original path planner also inadvertently generated some paths that are infeasible and impossible for a UAV to fly.

With flight mechanics introduced into the problem formulation as another constraint, the generated paths will be smoother with little to no sharp turns. This limits how close the newly generated paths can be to the original waypoints, reducing the potential for reconnaissance. Conversely, these changes will force the UAV to be further away from the threat, thus improving safety.

Five Scenarios Total Cost

Table 9 compares the average total cost for all scenarios tested to determine the overall impact of flight mechanics on the total cost of the path, reminder that smaller numbers are better. All components of the path (fuel, recon, safety, terrain, flight mechanics) are summed to arrive at a total cost of the path and averaged over all trials of a scenario. Comparing the total cost for the original PSO Path Planner and the new PSO Path Planner with flight mechanics will reveal whether the original path planner was returning feasible paths.

Table 9: Average total cost for three types of paths comparing with flight mechanics and without flight mechanics

Dominant Variable	Avg Fit w/o FM	Avg Fit w/ FM	% Increase
Scenario1			
Fuel	11.659	12.079	3.60%
Recon	23.774	25.055	5.39%
Safety	11.158	11.406	2.22%
Scenario2			
Fuel	11.033	12.118	9.83%
Recon	29.711	42.856	44.24%
Safety	11.960	14.127	18.11%
Scenario3			
Fuel	12.260	12.095	-1.34%
Recon	31.801	29.506	-7.22%
Safety	17.317	19.632	13.37%
Scenario4			
Fuel	12.112	12.461	2.88%
Recon	32.576	36.404	11.75%
Safety	12.274	12.609	2.73%
Scenario5			
Fuel	12.439	12.371	-0.55%
Recon	31.096	37.525	20.67%
Safety	12.509	12.308	-1.61%
Avg Total			
Fuel	11.900	12.225	2.72%
Recon	29.791	34.269	15.03%
Safety	13.044	14.016	7.46%

A general increase in the total cost of the paths is observed with the largest increase occurring in Scenario 2 with an increase of 44%. Some decreases in cost values were noted as well. Most were negligible changes of 1% or less, but Scenario 3 saw the reconnaissance paths decrease by 7%. This decrease can be attributed to the randomness that is inherent in PSO generating different results for the same problem every time it is run. With more trials of the same scenarios, the total cost

increases are expected to disappear. This is partially shown in the totals for all five scenarios, where there is an increase across all three types of paths.

Of particular importance is the relative magnitude of the changes for the three types of paths (fuel, recon, and safety). The changes seen in fuel efficiency and safety are both relatively small, on average 3% and 7% respectively. The changes seen in the reconnaissance paths are relatively large with an average increase of 15%. It can be concluded that flight mechanics has the largest impact on the reconnaissance factor of the paths, due to reconnaissance attempting to fly as close to targets as possible. This was often accomplished in the original path planner by making sharp turns in the flight path to move toward the target and then back way. Flight mechanics requires smoother turns, limiting the ease of which reconnaissance paths could be generated.

Real time path planning is desirable but not feasible for the evaluation of hundreds of alternative paths to find an optimal. All scenarios performed path planning at close to real time (approximately 5 seconds on average). The close to real time speed prove that basic flight mechanics can be incorporated into path planning without sacrificing computational time. As computer hardware advances, the amount of time required for path planning will go down and approach real time.

Path Visualization Results

Visually it is important to represent the paths to the operator in a clean and efficient manner for evaluation and selection. The original path planner returned paths with

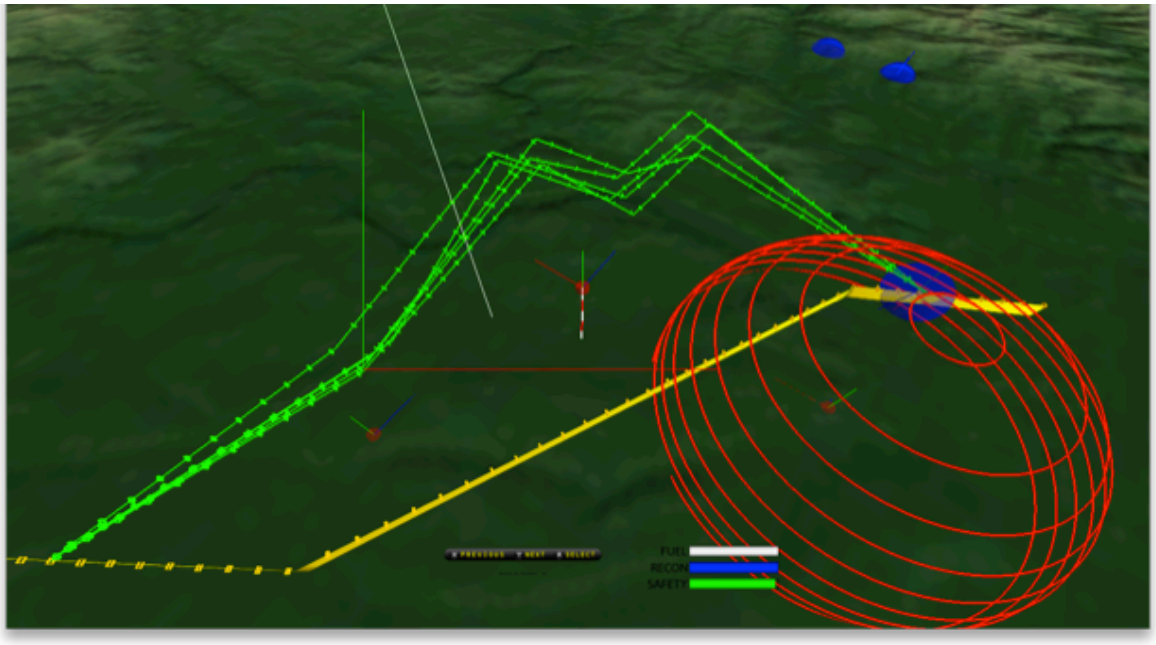

Figure 39: Original path planner visualization of alternate paths

linear interpolations between waypoints, as shown in Figure 39. While this method is efficient, it does not accurately represent the actual UAV path. The waypoints are chosen as equally spaced points along the b-spline curve calculated by PSO Path Planner. The new waypoints are not areas of interest nor do they denote characteristics of the flight path, such as a turning point.

Following the path exactly is impossible because it would require the UAV to make a physically impossible turn. The UAV instead follows a path close to the representation while traveling through the specified waypoints. Traveling through the waypoints and staying close to the defined path requires the UAV to navigate away from the path in order to provide enough room to make the required turn. Operators must reconstruct this path in their mind to understand the exact path the UAV will take. This process is mentally taxing as well as error prone due to operators making

assumptions about how the UAV will navigate the waypoints, such as when and how the UAV will turn.

B-spline interpolation allows for a smooth and accurate path representation that can account for the turning abilities of the UAV. Figure 40 shows the results of using a b-spline to represent the UAV flight path. Using universal interpolation, a b-spline representation of the UAV fight path is created using areas of interest as control points the path must travel through. The interpolation technique accounts for turns requiring a UAV to navigate away from the desired path in order to accommodate the UAV's minimum turn radius. A smooth path traveling through all the waypoints is generated. This representation is a more accurate representation of the path the

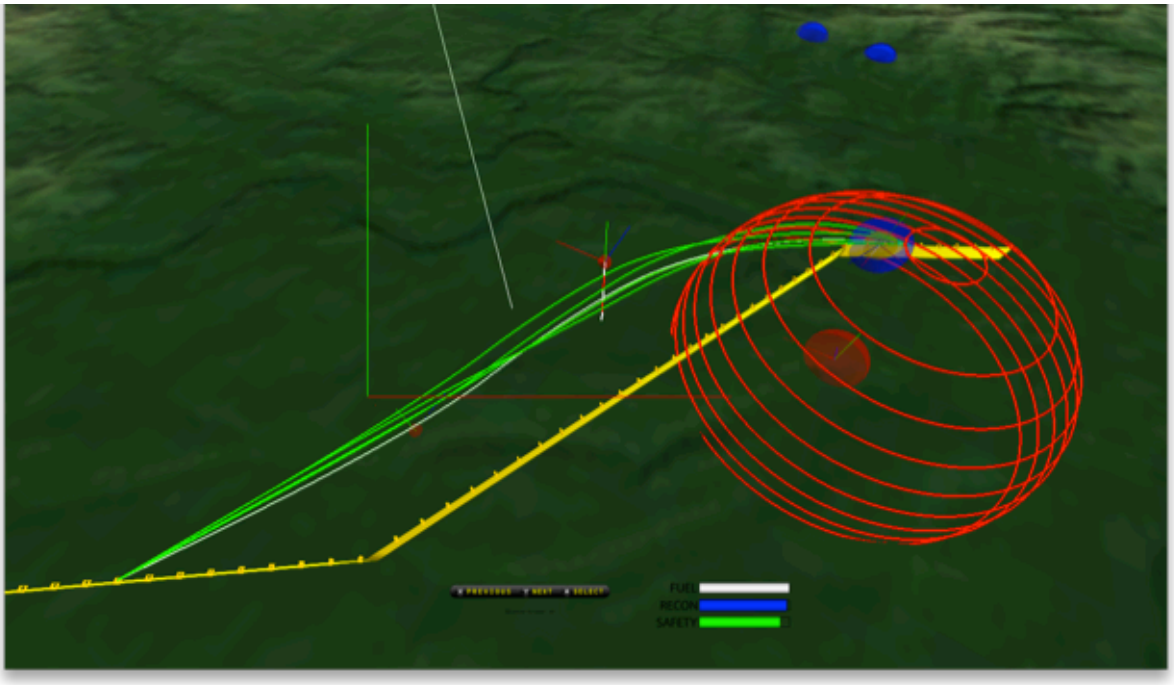

Figure 40: New path planner visualization of alternate path using B-Spline curves

UAV can fly than the linear interpolation of waypoints.

Visualizing the flight path as a b-spline curve saves the operator time and mental capacity by eliminating the need to mentally construct the UAV flight path from a series of waypoints. The representation more accurately represents the flight path of the UAV, which would benefit the operator and reduce the errors associated with mentally constructing the path. Reducing the number of tasks an operator must consciously think about can improve decision making and create a better operational environment.

Chapter 5 – Conclusions and Future Work

There were two primary goals of this research: to integrate flight mechanics equations into PSO Path Planner and create a better visualization of the flight paths. Changes to the path planner and the visualization of the flight paths provided both benefits and challenges. The benefits include improving the feasibility of the paths as well as the accuracy of the visualization. The challenges include eliminating all infeasible paths and increasing the fidelity of the flight models themselves.

PSO Path Planner Modifications

One limitation of this method is the flight characteristics are simplified for computational speed. The results prove that including flight characteristics in path planning impacts the feasibility of the paths. However, the models being used are greatly simplified from an aircraft's true performance characteristics. These models do not consider things such as side slip occurring when an aircraft makes a sharp turn. There are no considerations for thrust or drag. All of these things would provide a more complete model of the aircrafts flight characteristics and provide paths representing the complete flight capabilities of the aircraft.

The flight mechanics equations are used because they are computationally efficient but they do not yet allow path planning in real time because hundreds of paths must be evaluated to find an optimal. As computational hardware improves, the time required for path planning will decrease.

The equations used for determining the flight mechanics and the PSO problem formulation used allow the flight mechanics equations to be tailored to any type of aerial vehicle. All aircraft have certain static characteristics that can be used by the new path planner equations to generate the flight mechanics results. The new PSO Path Planner can be tailored to any aircraft by setting the weight, load factor, velocity range, and maximum altitude. This allows the path planner to be highly adaptable for many types of aircraft.

The single greatest benefit from the addition of flight mechanics to the path planning problem formulation is the reliability of returning feasible paths. The increase in total cost for the paths indicates the new path planner is returning more constrained paths due to flight mechanics. Increases in total cost across all three path types indicate the new path planner is returning feasible paths. Reconnaissance appears to be the most affected by the introduction of flight mechanics. This is due to the original path planner allowing sharp turns back towards areas of interest, where the new path planner requires smoother transitions.

The modifications made to the original path planner have shown the ability to achieve the goal of returning feasible flights. Just as important is the ability to calculate all the flight mechanics information in an online path planner. The implementation of flight mechanics in this path planner is still close to real time.

Flight Path Visualization

Switching visualization of the flight paths from a waypoint based system to a B-spline representation is no small change. The original PSO Path Planner utilized traditional b-spline curves for the analysis and used control points as waypoints. The new PSO Path Planner with flight mechanics incorporates universal interpolation to create flight paths traveling through the specified waypoints. This technique creates a more accurate representation of the flight path for both analysis and visualization.

Using b-splines to represent the flight path allows for a more accurate representation of the flight path to the operator. Waypoints require operators to mentally construct the flight path in their mind to determine how the aircraft would make turns. B-splines allow the representation of the turns through corresponding waypoints. This approach ideally will require the operator to spend less time mentally reconstructing the flight path.

The challenge is determining whether representing the flight paths using b-splines is more effective than the waypoint method. Curve representations are more information for the operator to process compared to linear paths. Flight paths are commonly visualized using waypoints. Proving the benefits of b-spline representation becomes a priority to confirm the hypothesis posed here.

Future Work

There are many areas of work that can be explored to further improve the PSO Path Planner. One issue is to explore the use of more complex flight dynamics. This work uses basic flight mechanics for a flight model. While these equations give an accurate representation of the flight characteristics, it would be interesting to explore how flight dynamics equations would impact the results.

A second issue is the presence of flight mechanics cost in some of the path planner results. It may be possible to increase the cost associated with violating the flight mechanics cost to eliminate any infeasible sections of paths. Increasing the complexity of the flight characteristics is also a possible solution. This would increase the computational time but would possibly increase the reliability of the results.

A third issue that could be investigated is the parallel processing of the PSO Path Planner. The Virtual Battlespace path planner runs PSO on a single thread. PSO is a process that lends itself well to parallelization due to the repetitive nature of the calculations over a large population. Running PSO on separate threads or utilizing the GPU for calculations would be an interesting avenue to pursue.

A four issue that could be addressed is inherent in the nature of PSO itself. During this research it was observed that the implementation of PSO with digital pheromones used might not provide the ideal functionality. This application is attempting to find fifteen unique solutions to a multi-modal problem. PSO is returning

paths that are in the same local minimum causing the path planner to find paths that are similar in location and shape. Running multiple runs of PSO for each of the desired fifteen paths would provide a larger selection of varied paths. This type of implementation would functionally provide results closer to the desired.

A fifth issue is the visualization of flight paths. This paper concluded that visualizing the flight paths using b-splines is a more accurate representation and saves the operator time. A user study should be conducted to determine if b-spline representations are the most effective method of visualization.

Acknowledgements

I would like to thank everyone who has helped me over the last two years. First, I would like to thank my parents Scott and Gerilyn who have provided me with love and support my entire life. You are both amazing role models and I would not be where I am without your guidance. Thank you to my sister Jamie who has constantly challenged me to be better and provided me with the encouragement to get there. Thank you to my entire family and friends for the support, encouragement, and laughter you have brought me over the last two years.

I would like to thank Eliot Winer, my major professor, who saw enough promise in me to take me on and guide me along the master's program. Thank you to my committee members Stephen Gilbert and Song Zhang for your encouragement and support of my research. Thank you to the staff at VRAC for always being available and willing to help in any situation.

Thank you to Eric Foo for helping me with research, answer questions, reading through my thesis, and all the little things you are always willing to do. Thank you to Brandon Newendorp and Kathleen Mettel for helping edit my thesis. A special thank you to Brett Nekolny, my partner in crime, the things we accomplished together and your friendship will remain with me forever. Finally, a thank you to all the members of my research group for making the last two years an unbelievably awesome experience for me.

Bibliography

[01] Barry, C.L., and Zimet, E., "UCAVs Technological, Policy, and Operational Challenges," *Defense Horizons*, Center for Technology and National Security Policy, National Defense University, No. 3, October 2001

[02] A Brief History, "Predator.jpg," http://abriefhistory.org/wp-content/uploads/2010/02/predator.jpg [Retrieved 26 May 2010].

[03] Engadet, "boeingphantom," http://www.blogcdn.com/www.engadget.com/media/2009/05/5-09-09boeingphantom.jpg, [Retrieved 26 May 2010].

[04] "Rise of the Machines: UAV Use Soars," Military.com, 2008, http://www.military.com/NewsContent/0,13319,159220,00.html, [Retrieved 26 May 2010].

[05] J. Clapper, J. Young, J. Cartwright, and J. Grimes, "Unmanned Systems Roadmap 2007-2032," 2007.

[06] members.cox.net, "Predator GCS," http://members.cox.net/renegade_sith2/miscjunk/predator-uav-controls.jpg, [Retrieved 26 May 2010].

[07] www.strategypage.com, "UAV vs. Airliner," http://www.strategypage.com/military_photos/200410111.aspx, [Retrieved 5 November 2010].

[08] Liggett, K.K., "Multi-UAV Supervisory Control Interface Technology (MUSCIT) Program", 2007.

[09] Miler, G.A., "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychological review*, vol. 63, pp. 81-97, 1956.

[10] Fuster, J., *The prefrontal cortex*, New York, Raven Press, 1980.

[11] Schallice, T., *From neuropsychology to mental structure*, Cambridge, Cambridge University Press, 1988.

[12] Just, M., Carpenter, P., "A capacity theory of comprehension: Individual differences in working memory," *Psychological review*, vol. 99, pp. 122-149, 1992.

[13] Miller, C., "Trust in Adaptive Automation: The Role of Etiquette in Tuning Trust via Analogic and Affective Method," *1st International Conference on Augmented Cognition*, Las Vegas, NV, pp. 22-27, 2004.

[14] Foo, J.L., Knutzon, J.S., Kalivarapu, V., Oliver, J.H., Winer, E., "Three-Dimensional Path Planning of Unmanned Aerial Vehicle in a Virtual Battlespace using B-Splines and Particle Swarm Optimization," *Journal of Aerospace Computing, Information, and Communication*, vol. 6, 2009.

[15] Brintaki, A.N., Nikolos, I.K., "Coordinated UAV Path Planning Using Differential Evolution," *International Journal*, vol. 5, pp. 487-502, 2005.

[16] Pongpunwattana, A., Rysdyk, R., "Real-Time Planning for Multiple Autonomous Vehicles in Dynamic Uncertain Environments," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, pp. 580-604, December 2004.

[17] Kabamba, P., Meerkov, S., Zeitz, F., "Optimal Path Planning for Unmanned Combat Aerial Vehicles to Defeat Radar Tracking," *Journal of Guidance, Control, and Dynamics*, vol. 29, pp. 279-288, March 2006.

[18] Hasircioglu, I.,Topcuoglu, H.R., and Ermis, M., "3-D path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms," *Proceedings of the 10th annual conference on Genetic and evolutionary computation - GECCO '08*, pp. 1499-1506, 2008.

[19] Hocaoglu, C. and Sanderson, C., "Planning multiple paths with evolutionary speciation," *IEEE Transactions on Evolutionary Computation*, vol. 5, pp. 169-191, June 2001.

[20] Mittal, S. and Deb, K., "Three-dimensional offline path planning for UAVs using multiobjective evolutionary algorithms," *2007 IEEE Congress on Evolutionary Computation*, pp. 3195-3202, September 2007.

[21] Nikolos, I.K., Zografos, E.S., and Brintaki, A.N., "UAV path planning using evolutionary algorithms," *Innovations in Intelligent Machines – 1*, Springer, pp. 77-111, 2007.

[22] Ruz, J.J., Arevalo, O., Pajares, G., and de la Cruz, J.M., "Decision making among alternative routes for UAVs in dynamic environments," *2007 IEEE Conference on Emerging Technologies & Factory Automation*, pp. 997-1004, September 2007.

[23] Kamrani, F., "Using on-line simulation in UAV path planning," *Simulation*, 2007.

[24] Kim, Y., Gu, D., and Postlethwaite, I., "Real-time path planning with limited information for autonomous unmanned air vehicles," *Automatica*, vol. 44, pp. 696-712, March 2008.

[25] Sinopoli, B., Micheli, M., Donato, G., and Koo, T.J., "Vision based navigation for an unmanned aerial vehicle," *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, pp. 1757-1764, 2001.

[26] Cooke, J.M., Zyda, M.J., Pratt, D.R., and Mcghee, R.B., "NPSNET: FLIGHT SIMULATION DYNAMIC MODELING USING QUATERNIONS," *Presence*, vol. 1, pp. 404-420, 1994.

[27] Jun, M. and D'Andrea, R., "Path planning for unmanned aerial vehicles in uncertain and adversarial," *Cooperative Control: Models, Applications and Algorithms*, 2002.

[28] Sands, W., Range, M., and Paso, E., "Drone Formation Control System Real-Time Path Planning," *AIAA Infotech and Aerospace Conference and Exhibit*, 2007.

[29] Antoniou, A., and Lu, W., "Practical Optimization: Algorithms and Engineering Applications," Springer, New York, 2007.

[30] Wikipedia.org, "Maximum Paraboloid," http://upload.wikimedia.org/wikipedia/commons/6/62/MaximumParaboloid.png, [Retrieved 29 May 2010].

[31] Mitchell, M., *An Introduction to Genetic Algorithms*, MIT Press, 1998.

[32] Kirkpatric, S., Gelatt, C.D., and Vecchi, M.P., "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671-680, 1983.

[33] Eberhart, R. and Kennedy, J., "A new optimizer using particle swarm theory," *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pp. 39-43, 1995.

[34] Lee, K.Y., and El-Sharkawi, M.A., *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*, Wiley-IEEE Press, 2008.

[35] Hassan, R., Cohanim, B., Weck, O.D., and Venter, G., "A COPMARISON OF PARTICLE SWARM OPTIMIZATION AND THE GENETIC ALGORITHM," *Proceedings of 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Austin, Texas: AIAA, pp. 18-21, 2005.

[36] Kalivarapu, V., Foo, J.L., and Winer, E., “Improving solution characteristics of Particle Swarm Optimization using digital pheromones,” *Structural and Multidisciplinary Optimization*, vol. 37, p. 415–427, 2009.

[37] Swartzentruber, L., Foo, J.L., and Winer, E., “Multi-Objective UAV Path Planning with Refined Reconnaissance and Threat Formulations,” *Proceedings of 6th AIAA Multidisciplinary Design Optimization Specialist Conference*, pp. 1-14, 2010.

[38] Piegl, L. and Tiller,W., *The NURBS book*, Springer Verlag, 1997.

[39] FalconView, “Falcon Features of FalconView Open Source,” [Retrieved 29 May 2010].

[40] Walter, B.E., Knutzon, J.S., Sannier, A.V., and Oliver, J.H., “Virtual UAV Ground Control Station,” *AIAA 3rd" Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, pp. 20–23, 2004.

[41] Microsoft Corporation, “Flight Simulator X,” http://www.microsoft.com/games/flightsimulatorx/screenshots.html#, [Retrieved 28 June 2010].

[42] Swartzentruber, L., Foo, J.L., and Winer, E.H., “Three-dimensional multi-objective uav path planner using meta-paths for decision making and visualization,” *AIAA*, pp. 1-16, 2008.

[43] Ruff, H.A., Calhoun, G.L., Draper, M.H., Fontejon, J.V., and Guilfoos, B.J., “EXPLORING AUTOMATION ISSUES IN SUPERVISORY CONTROL OF MULTIPLE UAVS,” *Proceedings of the Human Performance, Situation Awareness, and Automation Technology Conference*, pp. 218-222, 2004.

[44] Durbin, J. and Swan, I., others, “Battlefield visualization on the responsive workbench,” *Proceedings of the conference on Visualization’98, IEEE Computer Society Press*, p. 466, 1998.

[45] Hix, D., Swan II, J.E., Gabbard, J.L., McGee, M., Durbin, J., and King, T., “User-centered design and evaluation of a real-time battlefield visualization virtual environment,” *Proceedings IEEE Virtual Reality*, pp. 96-103, 1999.

[46] Guang, Y. and Kapila, V., “Optimal path planning for unmanned air vehicles with kinematic and tactical constraints,” *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 2, pp. 1301-1306, 2002.

[47] Scherer, S., Singh, S., Chamberlain, L., and Saripalli, S., “Flying fast and low among obstacles,” *Proceedings of the IEEE International Conference on*

Robotics and Automation, pp. 2023-2029, 2007.

[48] Langelaan, J. and Rock, S., "Towards Autonomous UAV Flight in Forests," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Fransisco, CA: 2005.

[49] Sanders, G. and Ray, T., "Optimal offline path planning of a fixed wing unmanned aerial vehicle (UAV) using an evolutionary algorithm," *IEEE Congress on Evolutionary Computation*, pp. 4410-4416, 2007.

[50] Kamrani, F. and Ayani, R., "Distributed Simulation and Real-Time Application," *Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications*, pp. 167-174, 2007.

[51] Jung, D., Ratti, J., and Tsiotras, P., "Real-time Implementation and Validation of a New Hierarchical Path Planning Scheme of UAVs via Hardware-in-the-Loop Simulation," *Journal of Intelligent and Robotic Systems*, vol. 54, pp. 163-181, July 2008.

[52] Garza, F.R. and Morelli, E.A., "A Collection of Nonlinear Aircraft Simulations in MATLAB," *NASA STI Program Office*, 2003.

[53] Schouwenaars, T., How, J., and Feron, E., "Receding Horizon Path Planning with Implicit Safety Guarantees," *Control*, pp. 5576-5581, 2004.

[54] Phillips, W.F., *Mechanics of flight*, John Wiley \& Sons Inc, 2004.

[55] Clancy, L., *Aerodynamics*, Halsted Press, 1975.

[56] Stengel, R.F., *Flight dynamics*, Princeton University Press Princeton, NJ, USA, 2004.

[57] Ceccarelli, N., Enright, J.J., Frazzoli, E., Rasmussen, S.J., and Schumacher, C.J., "Micro UAV Path Planning for Reconnaissance in Wind," *American Control Conference*, pp. 5310-5315, 2007.

[58] Lim, C., "A universal parametrization in B-spline curve and surface interpolation," *Computer Aided Geometric Design*, vol. 16, pp. 407-422, 1999.

[59] Kalivarapu, V. and Winer, E., "An Approach to Convert Vertex-Based 3D Representations to Combinatorial B-Splines for Real-Time Visual Collaboration," *43 rd AIAA Aerospace Sciences Meeting and Exhibit*, 2005.

[60] Shangming, W., Zefran, M., and DeCarlo, R.A., "Optimal control of robotic

systems with logical constraints: Application to UAV path planning," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 176-181, 2008.

[61] Hall, R., "Path Planning and Autonomous Navigation for use in Computer Generated Forces," *Combat Simulations*, FLSC, 2007.

[62] Soto, M., Nava, P.A., and Alvarado, L.E., "Drone Formation Control System Real-Time Path Planning," *AIAA, Infotech@ Aerospace 2007 Conference and Exhibit*, 2007.

[63] "X-45 J-UCAV Joint Unmanned Combat Air System, USA," http://www.airforce-technology.com/projects/x-45-ucav/specs.html, [Retrieved 3 November 2010].

[64] "General Atomics MQ-1 Predator," http://en.wikipedia.org/wiki/General_Atomics_MQ-1_Predator, [Retrieved 3 November 2010].