UAS SAFETY PLANNING AND CONTINGENCY ASSESSMENT AND ADVISORY RESEACH

John DiFelici and Chris Wargo, Mosaic ATM, Leesburg, VA

Abstract

Operation of Unmanned Aircraft Systems (UAS) require the same degree of pre-flight and mission planning as that preformed for manned aircraft. The goal of routine operations has an evolving safety case which must include the review of actions and conditions of the aircraft and the pilot during normal, off-nominal, abnormal and emergency events. UAS flight planning is currently a manually intensive process, and trying to replicate a pilot's decision making course of actions for mission contingencies is an exceptionally tedious and lengthy activity. Additionally, the incorporation of contingency/emergency routes is left up to the planner and there is no guarantee of sufficiency or consistency of contingency plans when activated.

To support the recurring assessment of these actions, Mosaic ATM has performed a 2 year NASA project to development a mission safety assessment and contingency support tool called Aviate. Our project objective is to automate the UA contingency flight planning process providing uniform accounting to known conditions, hazards, and other factors. We will field, increasingly, a capable product suite that supports the mission planner, the air traffic controller, the PIC, and the onboard intelligent piloting function. This will provide a level of confidence and an equivalent level of safety to unmanned aircraft mission planning prior to any UAS departures from airfields in the NAS or flights into through the NAS.

Introduction

This paper reports on the research and development of a UA flight plan safety assessment and flight path / position contingency support tool called Aviate. The Aviate system provides a set of tools and algorithms for checking the safety of UA flight paths and for finding and safety checking contingency routes to local landing locations. The Aviate system ingests information from multiple data sources: population densities, controlled airspace, geospatial / urban development data, air traffic density, aircraft performance models, convective weather, surface winds and winds aloft, and real-time airspace traffic information. Aviate uses this data in order to define the risk, compute the risk, and minimize the risk of UA flight plans and contingency routes.

The first portion of the Aviate project was to research and develop a set of systems requirements. To document the analysis, the project team selected to use Goal Structured Notation (GSN) to capture the argument threads within UAS operation. This provided use cases and functional requirements that were then developed into a prototype toolkit.

The system algorithms are a series of safety checks across the entire route of flight. This tool can be used in a research desktop environment or in the mission planning phase. During operation the system can be used in real-time to provide informational options to the pilot in command or to air traffic controllers. Mosaic ATM has completed two releases of functionality under the NASA project.

This paper describes the project requirements development using the GSN approach, and provides a table of safety checks used for algorithm development. A-Star and Dubins search path algorithms exist, but have not been combined or applied commercially to UA flight and contingency route planning. The Aviate innovation brings together these two path planning algorithms, GSN based decision making, and several static and real-time data sources to provide a UAS contingency path planning and scoring system.

First, an overview of the Aviate system is presented, followed by a technical review of the research and algorithms used within Aviate. Potential future development of the system is then discussed, followed by a closing summary.

Aviate Overview

Aviate is a set of tools, analyses, and algorithms used together to create a system for scoring UA flight plans, generating contingency routes for points along a UA flight plans, and then scoring the generated

978-1-5090-2149-9/16/$31.00 ©2016 IEEE

routes according to parameters defined by the user. Each contingency route is optimized for avoiding known obstructions while minimizing route length and then scored according to user provided preferences. The algorithm compiles ground and airspace data (e.g. terrain, urban structures, critical infrastructure, airports, runways, ditch sites, airspace class, FCA, Wind, METAR, TAF, etc.) to build a cost surface and utilizes a hybrid A-Star path finding algorithm with a Dubins path analytic extension for path termination to generate a set of contingency routes. It then scores each route according to a set of rules and parameters defined by the user.

The Aviate system was envisioned and designed as a product with increasing incremental functionality to perform safety and contingency checks on UAS operations in a variety of operational settings and modes. The list of use cases and operation settings includes (but is not limited to) the following:

Case 1: Static Safety Assessment Tool

This use case is mainly for researchers and/or safety analysts. The data used in this case is all static or historical and any changes made to the system state parameters are done by the user. The tool, in this mode, provides:

- User configured UA 'missions' which define: UA specifications, flight path scoring rules, and contingency path route planning parameters.
- Visualization of the area of interest, the flight path, scoring and contingency path anomalies.
- Users can score flight paths for defined missions.
- Users can automatically generate contingency routes along the entire flight path.
- Support the use of multiple users with public/private and update/delete data permissions where appropriate.

Case 2: FAA Flight Planning and Approval Tool

In this use case, the tool acts as an aid to the Certification of Authorization process (under the Mosaic ATM defined approach of "File N' Fly" in the NAS) with real-time approval and support for real-time operations. The data and configuration information is the same as the UAS Contingency Advisor Tool. The tool, in this mode, would not have any visible display. It would instead, score submitted UA flight plans using static and real-time data, and return a Go or No-Go for the flight plan to the submitter.

Case 3: Safety Assessment Module

In this mode of operation, the tool is constructed as a module or a backend service (can be local or remote) that is used by another system, such as NASA's LVC-DE simulator, to score flight plans and to automatically generate contingency routes.

Case 4: UAS ATC Contingency Advisor Tool

This mode of operation is designed to be used in an ATC environment with real-time integration into FAA systems. The static data used by the tool (UA specifications, contingency route parameters) is predetermined or redefined on a periodic update cycle. Real-time data (Monitor Alert/MAP values, sector loading, and UAS position updates) come from integration with FAA systems and the UAS controlling system. The tool, in this mode, provides:

- An appropriate display which would be integrated into existing FAA equipment.
- Real-time updates to UA state or external data changes (like weather forecast updates) are accounted for and may cause updates to contingency route planning.
- The controller can have the tool make suggestions (a ranked list) for contingency routes for off-nominal events at any point along the flight path.

Case 5: UAS Pilot Contingency Advisor Tool

This mode of operation is designed to be used in by a UAC Pilot with real-time integration or data access to the UAS system. The system would be provided real-time UA status and environmental data and could provide the pilot with updated safety assessments of the current and future portions of the flight plan, in addition to providing contingency options for the UA at its current position.

Case 6: On-board Contingency Advisor Tool

This mode of operation is designed to be used as an integrated piece of software into the UA avionics components. The system could be called upon to provide contingency routes for an aircraft experiencing problems during its flight. Depending upon the type of trouble, contingency flight plans of varying safety levels could be provided to the UA for landing options.

Architecture

The Aviate system architecture was developed to be a client / server architecture in order to satisfy the following objectives:

- Basic algorithmic functionality should support multiple client types (i.e. clients for the various use cases mentioned above).
- Support for communications with external systems (simulators, telemetry systems, etc.)
- Computational tasks can be farmed out, if necessary, to multiple hosts.
- Data sharing between users: UA definitions, missions, scoring parameters, etc.

Figure 1 depicts a model of the system architecture comprised of four main components: the Aviate Server which contains the business logic, the Job Engine which handles the algorithmic computations, the Data Services which feeds data to all the modules in addition to ingesting real-time data (weather, air traffic, etc.), and the Aviate clients which cover some of the use cases above.

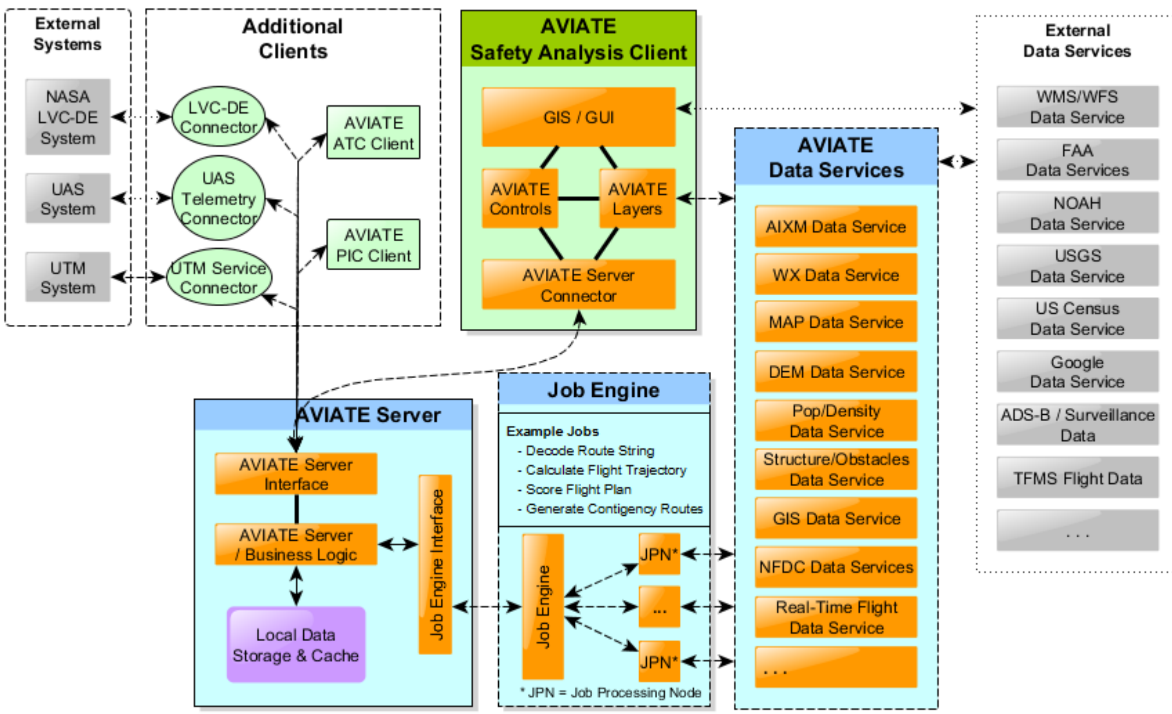

Figure 1. AVIATE Architecture

Currently the Aviate system, over the course of the SBIR Phase 1 and 2 projects, has been developed to cover use case 1 (the Safety Assessment tool via a software client), use case 3 (via an API), and use case 4 (via a software client). Development for cases 2, 5, and 6 has been deferred until a future date. The next two sections provide a brief overview of the Safety Assessment client and the ATC Contingency Advisor client.

Safety Assessment Tool (Use Case 1)

The Safety Assessment tool is one of the two clients released with the current Aviate system. Briefly, it provides several screens allowing the users to define UA missions, UA Specifications, path finding parameters, path scoring parameters, and UA landing locations. Figure 2 depicts the data management window.

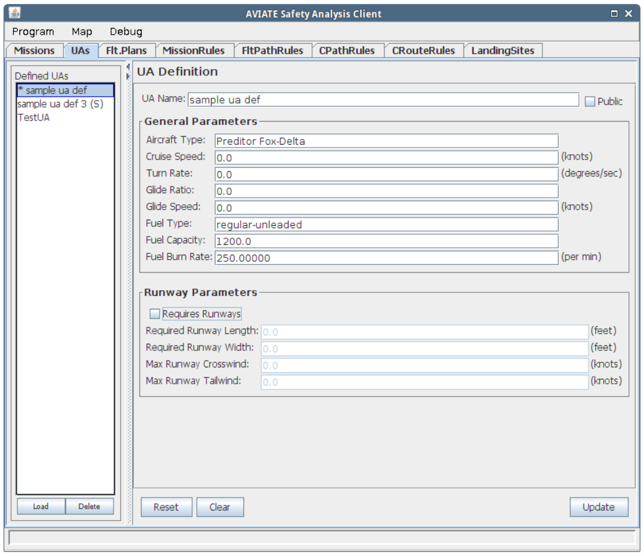

Figure 2 . Safety Client Data Management

The client allows users to submit UA missions to the server to score flight paths and find contingency routes along the flight paths (see Figure 3), and to also compute contingency routes from hand picks locations (see Figure 4).

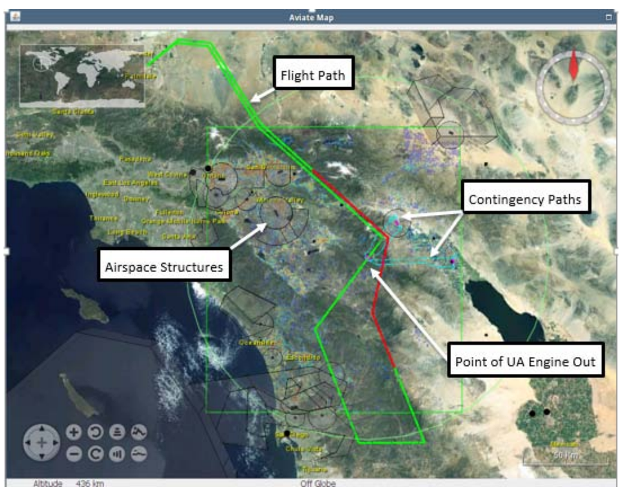

Figure 3. UA Mission Flight Path Scoring

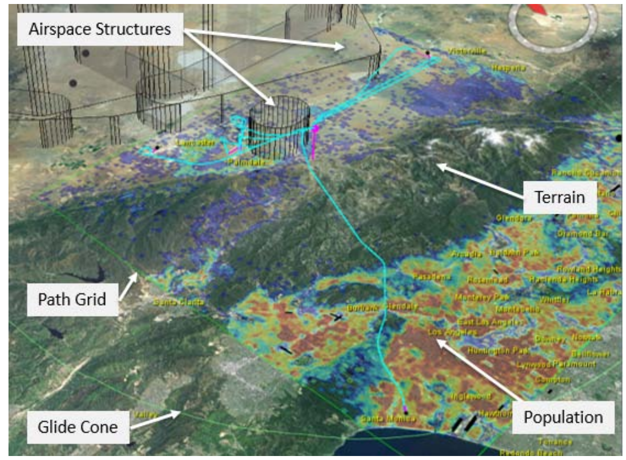

Figure 4. Contingency Route Generation

ATC Contingency Advisor Tool (Use Case 4)

The ATC Contingency Advisor tool is the second of the two clients released with the current Aviate system. The client itself consists of a simple dialog that displays: the names of the current active UA flights in addition to any flights in the nearby vicinity, an area for a controller to submit a contingency route for a specified UA flight, and finally, a list of computed contingency routes and landing locations generated and scored by the Aviate safety algorithms (see Figure 5). The developed client was kept simple due to the fact that if it is ever to be incorporated into an actual ATC display, it would need to small and take up minimal space.

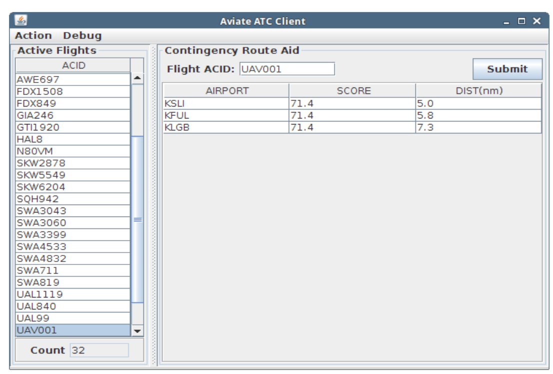

Figure 5. ATC Contingency Advisor

Unlike the Safety Assessment Client which uses only static and historical data, the ATC Contingency Advisor tells the server to utilize the real-time data collected by the system to account for updated weather and nearby air traffic congestion. Thus, contingency routes generated for the current UA positions will avoid the usual static obstacles, in addition, to staying clear of current air traffic. Figure 6 shows a picture of some real-time flight traffic, displayed in the Safety Assessment client for purposes of visualizing the real-time flights. The active flights are colored in blue with blue pins depicting their current location along their planned flight path.

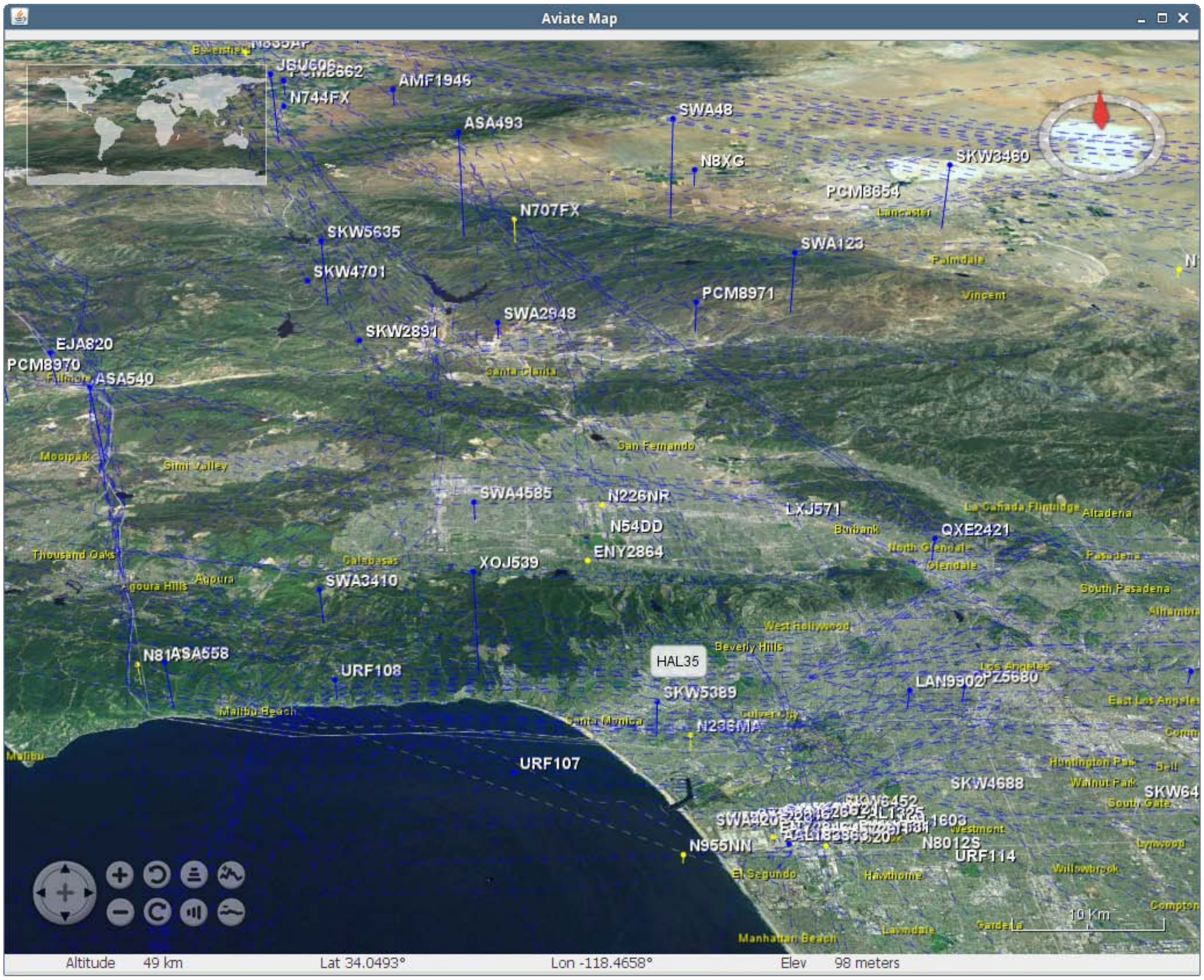

Figure 6. Real-time Flight Tracks and Positions

The next three sections discuss the safety case analysis and development goals using the GSN approach, the path finding algorithms, and the flight plan and contingency route scoring methodology.

GSN

Aviate algorithms are linked to the assessment of the UAS safety case developed using an analysis method known as Goal Structuring Notation (GSN). At this stage in the development of the UAS integration into the NAS, there is not a community consensus for the safety case(s) covering UAS operations in the NAS. At best, with the publishing of the RTCA DO-344 [1], there is now a current assessment for the qualitative safety objectives for each of the major ORs and FRs of the UAS. However, these safety objectives, developed by a fault hazard assessment method, are mostly component focused. That is, they deal with the failure, or erroneous operation of the functions performed by a component and its impact upon normal behaviors. These are not necessary related, or traceable, to the operational behaviors of the UAS when flown in a selected mission type. The RTCA DO-344 work did not cover other UAS functions accept those related to the aircraft avionics. To understand the arguments and evidence needed to assess operations under specific use cases (mission types), the GSN method is a very useful approach [2, 3].

GSN employs a graphical decomposition of safety goals that can be traced and reviewed. This allows for enhanced communications between engineers and users of the safety case. This can facilitate an understanding and acceptance across a range of disciplines (e.g., engineering, operations and managers). GSN provides a set of symbols to structure the safety case using its goals, arguments flows, strategy of assessment and evidence. The basic symbols for doing so are shown in Figure 7.

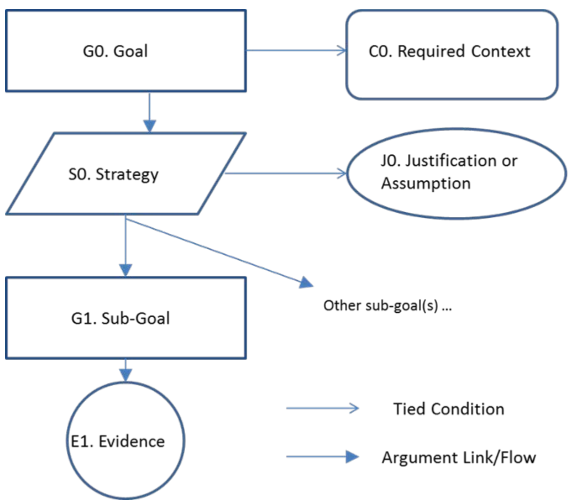

Figure 7. Basic GSN Symbols

The first task in the GSN method is to define the top goals. This involves understanding the decomposition of the UAS Use Cases into a UAS Safety Case for a selected mission. For Aviate, we separated the UAS operations into normal and non-normal. In aviation, this separation is often confusing because there are operational events, scenarios and conditions that are clearly not normal, but for which the systems, pilots, and controllers must deal with on a routine basis. These non-routine events are often classified as "common off-nominal." Figure 8 provides a contextual view of this discussion of classifying events and/or scenarios that make up a safety case.

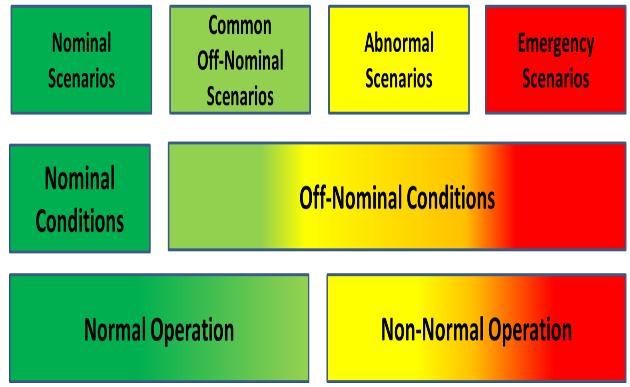

Figure 8. Mapping Context for Operations

The GSN graph for Aviate links its software functions with the four groupings of system use scenarios (or events) as shown in the top row of Figure 8: Nominal, Common Off-Nominal, Abnormal, and Emergency. This provides an explicit mapping of how the use of Aviate can be traced to solving a safety assessment. In order to more easily categorize solutions, the *evidence bubbles* (as functions of Aviate, Pilot/UAS requirements, FAA requirements, or no solutions available) were color coded as seen in Figure 9.

Figure 9. Color Coding for Aviate GSN Graph

Figure 10 *(full sized PDF available upon request)* depicts the entire Aviate GSN analysis graph with seven top level goals: G1 – G7. Goals G1 – G3 are used to direct graph flow to one of the four main branches: goals G4 for Nominal operations, G5 for Off-Nominal operations, G6 for Abnormal operations, and G7 for Emergency operations. Table 1 provides a brief description of all the terminal goals (i.e. those goals which are not broken into further sub-goals) and categorizes each goal by its applicability to either the UAS aircraft itself, the overall mission, or N/A for not-applicable for Aviate scoring.

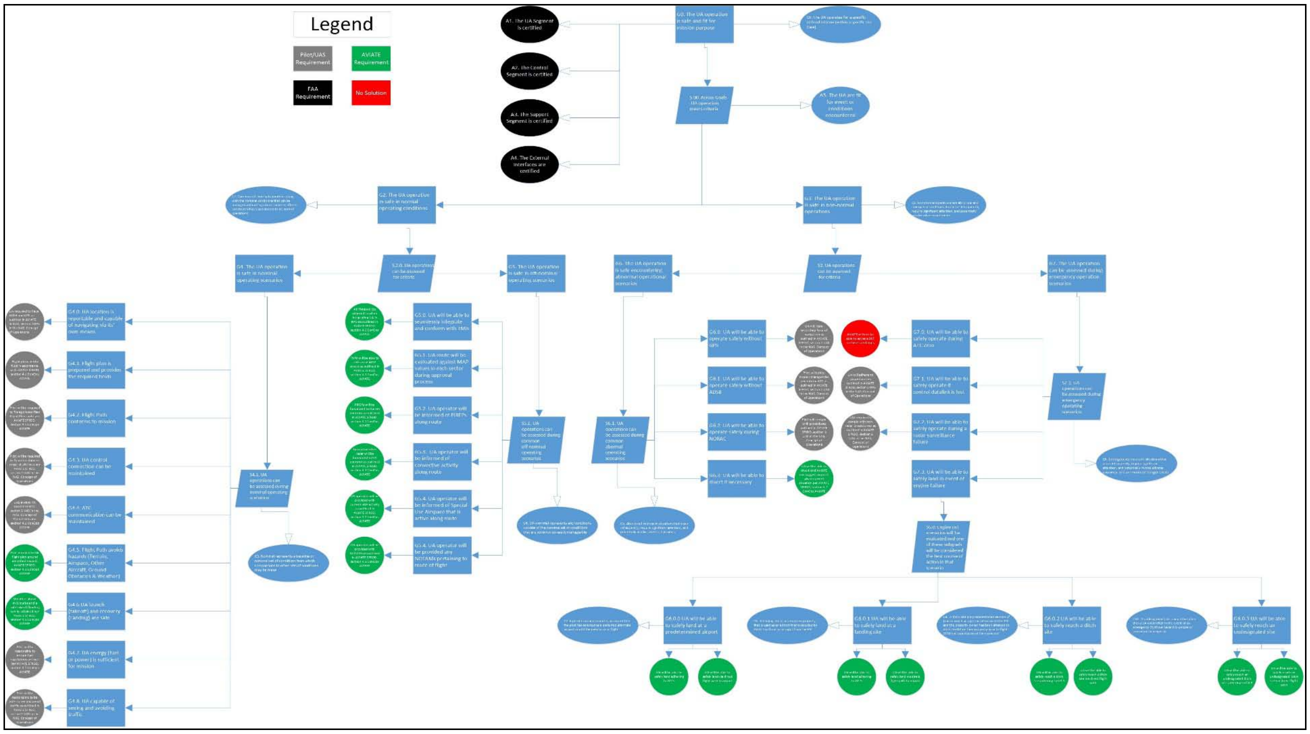

Figure 10. Full Aviate GSN Graph

Table 1. GSN Scoring Goals

Goal	Description	Category
G4.0	UAV location is reportable and capable of navigating via its' own means.	UAS
G4.1	Flight plan is prepared and provides the required fields.	UAS
G4.2	Flight Path conforms to mission.	UAS
G4.3	UAV control connection can be maintained.	UAS
G4.4	ATC communication can be maintained.	N/A
G4.5	Flight path avoids hazards: terrain, man-made obstacles, airspace, other aircraft, weather, etc.	Mission
G4.6	UAV launch and recovery are safe.	Mission
G4.7	UAV energy (fuel or power) is sufficient for mission.	UAS
G4.8	UAV capable of seeing and avoiding traffic.	UAS
G5.0	UA will be able to seamlessly integrate and conform to TMIs.	Mission
G5.1	UAV route will be evaluated against MAP values in each sector during approval process.	Mission
G5.2	UAS operator will be informed of PIREPs along route.	N/A
G5.3	UAS operator will be informed of convective activity along route.	N/A
G5.4	UAS operator will be informed of Special Use Airspace that is active along route.	N/A
G5.5	UAS operator will be provided any NOTAMs pertaining to route of flight.	N/A
G6.0	UAV will be able to operate safely without GPS.	UAS
G6.1	UAV will be able to operate safely without ADSB.	UAS
G6.2	UAV will be able to operate safely during NORAC.	UAS
G6.3	UAV will be able to divert if necessary.	Mission
G7.1	UAV will be able to safely operate if control data-link is lost.	UAS

G7.2	UAV will be able to safely operate during a radar surveillance failure.	N/A
G7.3.0	UAV will be able to safely land at a predetermined airport.	Mission
G7.3.1	UAV will be able to safely land at a landing site.	Mission
G7.3.2	UAV will be able to safely reach a ditch site.	Mission
G7.3.3	UAV will be able to safely reach an undesignated site.	Mission

Contingency Path Algorithm

The Aviate contingency path finding algorithm combines graph planning with continuous state representations and methods for expediting / accelerating solutions. The algorithm is built from what is called a Hybrid-State A* algorithm, which consists of a Generic A* algorithm with analytic extensions for path completion. Guarantees are provided on the "realizability" of solutions by the physical system under consideration, which in this case is an aircraft [4, 5]. Notions of the "cost" for an aircraft occupying a region of airspace are part of the formulation as is the concept of "cost to go" to the goal, which is often either a physical location denoted by the geodetic coordinate or an arc or loxodrome in the sky that defines a flight plan segment. Constraints on regions of penetrable airspace, altitude constraints, speed constraints, turn-rate constraints, and such things as the desire to maintain safe spacing away from the boundary of 'obstacles' all fit nicely within this framework.

Given a starting position (latitude, longitude, altitude, and heading), the algorithm first computes the glide cone for the UA using the UA's glide slope parameter and then queries all the airports and user defined landing sites and ditch sites within the glide cone. All the landing sites are checked to see whether their elevations are above or below the glide cone, and any airports above the cone are removed from consideration. For UAVs requiring runways, the remaining airports have their runways queried and checked against the required UA runway parameters; runway length and width must be within UA requirements. Additionally, if wind information is available for the airport, crosswind and tailwind information is computed for each runway and checked against UA tolerances. The runways (and runway directions) that fall outside the UA allowable crosswind and tailwind parameters are removed from consideration. Lastly, for all the remaining landing sites, a simple path is computed from the starting position to each runway end using a Dubins Path algorithm [6], and those runway ends with path lengths greater than the allowable glide distance are rejected.

Once a set of potential path ending locations have been identified, search end-point goals and a search space are created for the path-finding algorithm. The end-point goals are created by taking a one-nautical mile point off of each valid runway end at the appropriate glide-slope altitude above the runway. End goals for user defined landing sites and ditch sites do not require the alignment segment. The search space is then created by using the bounding box containing all the end-point goals and then extending each side by an additional distance equal to three times the turn radius of the UA.

With all the search algorithm inputs now created, the path generator iterates over all of the end-point goals using the same starting position and computes (or fails to compute) a path to each goal using the generic A* algorithm with Aviate-specific procedures. The remaining sub-sections provide more details on the algorithm specifics. The results of each start-to-goal route, either a success or failure, are saved and sent back to the Aviate program for final scoring using the Contingency Route Scoring algorithm.

Generic A Algorithm*

The generic A* algorithm used by Aviate is based upon the A* formulation as defined in [7]. It uses a general Dykstra search algorithm with both path cost and distance-to-goal costs (which account for the A* variation), and all the specific cost and test functions of the algorithm are contained within the Aviate-specific formulation of the search problem. Figure 11 shows an overview of the algorithm process.

The generic A* algorithm is initialized with the starting position, end-state goal position, and a time limit. Additionally, a set of problem-specific functions are provided to the algorithm that tune the generic algorithm to the specific problem trying to be solved. (The Aviate-specific functions are denoted with the prefix 'AV_', and the inner workings of each function will be detailed in the next sub-section.)

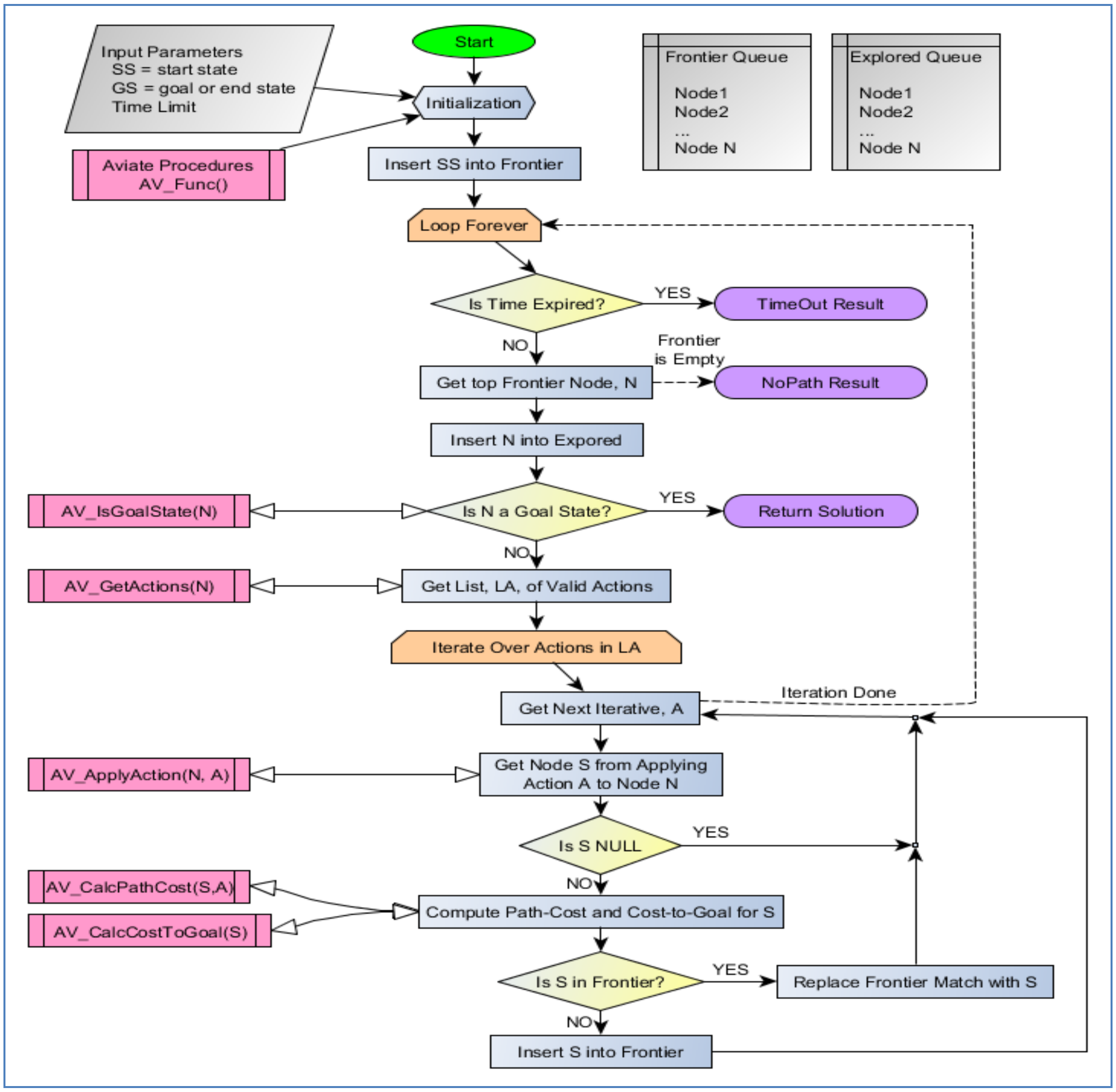

Figure 11. Generic A* Algorithm

To begin, the algorithm adds the starting position node to the Frontier queue. The Frontier queue contains nodes sorted by their total path cost in ascending order. The top of the queue contains the node with the least path cost. The algorithm then goes into an infinite loop in which it only exits if 1) the specified time limit for the algorithm has been reached, 2) there are no more nodes in the Frontier queue, or 3) a node has reached the goal state. The first two of these exit conditions return a non-solution result, while the last exit condition returns a result with a solution.

Inside the loop, the algorithm pulls off the top node from the Frontier queue and puts it into the Explored queue, which contains a list of all the nodes that have been explored. It then checks whether the current node is equivalent to the goal state by calling the specialized Aviate function AV_IsGoalState. If 'true', then we have found a path from the start to the end-point goal and can return a solution. Otherwise, we call the AV_GetActions to get a list of all possible actions from the current node.

Next, the algorithm iterates over all the actions, and for each action, a new child node is created by applying the current action to the current node through a call to the function AV_ApplyAction. If a NULL result (no node) is returned from the function, then the algorithm iterates to the next node in the list. Otherwise, the path cost and the cost to goal are computed for the child node via the AV_CalcPathCost and AV_CalcCostToGoal functions.

Finally, the child node is placed within the Frontier queue. If the equivalent state of the child node is found within the Frontier queue, then the current child will replace the equivalent in the queue rather than just be inserted.

AVIATE Specific Procedures

This section provides a brief look at the Aviate-specific functions used by the generic A* algorithm.

AV_IsGoalState: This function checks whether the current node is equivalent to the goal node (to within some user-defined tolerance), and if so, it returns a "true" value. If not, and if the distance between the current node and the goal is within a user-defined analytical expansion distance parameter, then the function will try to calculate an analytical path from the current node to the goal. The attempt at the analytical solution is a result of the kinematic constraints on the search path.

Under many circumstances, a generic Dyktra algorithm (or variant) will always find a path (if one exists) between the starting and end points. However, because our problem deals with a physical aircraft subject to real physical constraints and kinematics, the solutions found by a general algorithm may not be physically possible. Additionally, if the general algorithm has physical constraints placed upon it (as is done for this Aviate problem), then a solution is not guaranteed, or the solution times may be greatly increased over a non-constrained case.

Therefore, for the Aviate problem, this function allows for the attempt of an analytic solution if the current node is a user-specified distance from the goal position. The analytical solution calculated is the minimum distance solution from either an external tangent Dubins Path or an internal tangent Dubins Path—see the Dubins Path sub-section below for more details on the Dubins Path calculations. If a solution exists and is returned, the analytical path is converted into a set of nodes and assigned path costs. If the path-to-node conversion process fails (due to out-of-bounds conditions, altitude, or terrain conditions, etc.) then a "false" is returned. Otherwise, this function returns a "true" because the analytic solution can be used to take the UA from the current node to the goal.

AV_GetActions: This function returns a list of actions available to the algorithm from the current node state. The three actions available in the Aviate problem are 1) turn left, 2) turn right, and 3) continue straight. If the current node action is left turn, then the returned actions are to continue the left turn and to fly straight. If the current node action is right turn, then the returned actions are to continue turning right and to fly straight. If the current node action is to fly straight ahead, then the returned actions are to turn left, turn right, and to continue flying straight.

AV_ApplyAction: This function creates a new node by applying the current action to the current node. It applies the turn logic or fly straight logic to the current node and creates a new node at a calculated distance furthering the flight path. Checks are performed on the new node to ensure that it's within the search space and has not gone below the terrain elevation. If any of the checks fail, a NULL result is returned instead of the newly created node.

AV_CalcPathCost: This function calculates the path cost of the current node by summing up the values of several cost maps. The cost maps include many user-defined cost factors in addition to using the current node state and goal state to create a cost value.

AV_CalcCostToGoal: This function calculates a cost for getting from the current node to the goal node. The calculated cost is a cost-to-goal scale factor times the larger of 1) the distance between the current node and the goal node and 2) the arc length of the UA turn through an angle, which is the difference between the heading required by the goal node and the heading of the current node. This equation applies pressure to get closer to the goal node when far away and to then get into alignment with the goal heading when closer to the goal node.

Dubins Path Algorithm

The Dubins Path algorithm, used for analytically expanding the final portion of the contingency path solutions, allows one to find an optimal path between two points and headings using only left turns ‘L’, right turns ‘R’, and straight segments ‘S’, assuming that a path can be found given physical constraints. All valid paths will consist of one or more of the following actions: RSR, RSL, LSR, LSL, RLR, LRL, LS, RS, etc. Two Dubins Path algorithms are used in the AVIATE program—an ‘external’ tangent algorithm and an ‘internal’ tangent algorithm—and the one that produces the shortest path is used for the solution.

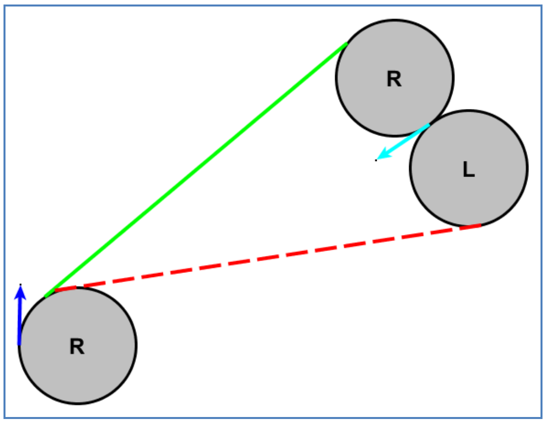

Figure 12. Dubins Path Example

Figure 12 shows an example of both an external tangent path and an internal tangent path. The dark blue arrow is the starting position, and the light blue arrow is the ending position. The gray circles represent ‘R’ right turns and ‘L’ left turns. The green tangent line is the external tangent connecting two right turns for a RSR solution, and the dashed red line is the internal tangent connecting a right and left turn for a RSL solution.

Flight Path Scoring

The scoring for the both the contingency routes and the overall flight paths follow the same scoring methodology. In order to score routes in a manner that is meaningful and relevant to the user, a set of germane attributes needed to be defined and utilized by the route scoring algorithm. The attributes that were identified were the various types of landing areas, the types of wind at the landing locations, use or non-use of Special Use Airspace (SUA), and whether the flight would encounter any convective activity along the route.

GSN Scoring Criteria

Aviate uses the GSN graph, see Figure 10, and terminal end goals, see Table 1, as a basis for the flight path scoring algorithm criteria. In addition to the scoring criteria, default (suggested) weighting factor(s) are provided.

Scoring Algorithm Attributes

The criteria and goals listed in this section apply to UAS missions; and while suggested weighting factors are detailed, the actual weighting factors provided by the user for real or specific UAS missions may vary greatly from mission to mission. The scoring algorithm uses the contributions for all of the attributes described below, normalizes them to a total score of 100, and provides a score back to the user from 100 (a perfect score) down to the minimum near zero for routes / paths that contain less than optimal segments.

The subsections below specify the various scoring attributes, and Table 2 shows the default scoring values for each attribute or attributes set, though the user is capable of altering these values to whatever they feel is appropriate.

Table 2. Scoring Attribute Default Scores

Landing Area	**Score**
Airport	45
Landing Site	30
Ditch Site	15
Undesignated Ditch Site	5
Wind	**Score**
Crosswind	15
Tailwind	15
SUAs	**Score**
No SUA	5
Convective Weather	**Score**
None	20
Moderate	15
Heavy	5
Extreme	0
Airspace Class	**Score**
None	15
Class B-E	5
Class A	0
Energy	**Score**
Sufficient Energy	10
Complete Flight Plan	**Score**
Completed Flight Plan	5
Control Connection	**Score**
Control Connection	5
Safe Launch, Recovery	**Score**
Safe Launch & Recovery	5
TMI Integration & Conformity	**Score**
Conformity & Integration	5
Flight Plan Adherence to Sector Approval and MAP Evaluation	**Score**
Adherence to Approval	5
Location & Navigation	**Score**
Loc. And Nav. Good	5
Sense and Avoid	**Score**
Sense and Avoid	5
GPS Free Operation	**Score**
GPS Operation	5
ADSB Free Operation	**Score**
ADSB Operation	5
NORAC Safe Operation	**Score**
NORAC Safe Operation	5
Contingency Routes Available	**Score**
Contingency Routes (S_{CR})	10

Landing Areas

Landing areas consist of Airports, Landing Sites, Ditch sites, and Undesignated Ditch sites and numerical scores are associated with each type of landing site. A flight path attains its maximum site score if it is landing at an airport / or user defined landing site at which it could make the runway (for airports) and had the appropriate final approach course. If less than a specified final approach is available, but more than 25% is available, then the formula below is used to compute a partial score – for less than 25% a score of 0 is given.

$$S_P = (D_A / D_S) * S_F \quad (1)$$

S_P is the partial score, S_F is the full score, D_A is the available distance, and D_S is the user specified desired distance.

Winds

The maximum tolerable crosswind and tailwind speeds are provided by the pilot when filing the flight plan, and are used by the wind scoring algorithm. The crosswind and tailwind scores are scored according to the following formula:

$$S_P = (1 – F) * S_F,\ F = W_M / W_A \quad (2)$$

S_P is the partial score, S_F is the full score, W_M is the measured wind (tailwind or crosswind) and W_A is the maximum allowed wind as specified in the flight plan.

Airspace Structures

The Airspace SUA criteria checks if the flight path avoids SUAs. If the flight path remains outside all SUAs then it will receive the full SUA score; otherwise the plan will get a zero for this criteria.

Convective Weather

This criteria checks if the flight plan comes within 15 nm of any convective activity. If the flight plan does not encounter any convective activity then the "No Convection" score is awarded. If the flight plan does travel within 15nm of convective activity, then the awarded score will be from one of the "Moderate", "Heavy", or "Extreme" categories. If multiple convective areas are intruded upon, only the score from the convective spot with the most severe category will be awarded.

Airspace Class

This criterion checks the flight plan against intrusions into unqualified classes of airspace. If the flight plan stays within its qualified airspace class(es) it receives the full score. If and unqualified intrusion occurs, the score from the highest class airspace (A highest, E lowest) is used.

UAS Energy

This criterion is used to indicate that the UAS has sufficient energy (fuel or power) to complete its mission. The criterion is checked by performing a rough calculation using the flight plan distance, energy capacity, and energy usage/consumption rate. If the total energy consumed (using the consumption rate) over the entire flight plan distance is less than or equal to the total energy capacity of the UA plus and required reserve, then the full score is given. If the total consumption is less than the UA energy storage plus required reserve but greater than just the UA energy storage, then the score will be a partial score according to the formula below. If the energy consumed is less than the energy stored, then a score of zero results.

$$S_P = S_F * (E_U - E_S) / R, \tag{3}$$

S_P is the partial score, S_F is the full score, E_U is the energy used, E_S is the energy stored, and R is the required reserve.

Complete Flight Plan

This criterion is used to indicate that the flight plan for the mission has been properly prepared, provides all the required fields, and conforms to any mission requirements. If checked this criteria provides a full score.

UAS Control Connection

This criterion is used to indicate that the UAS control connection can be maintained throughout the entire mission. If checked this criteria provides a full score.

Safe UAV Launch & Recovery

This criterion is used to indicate if the launching and recovery of the UAV is considered "safe". If checked this criteria provides a full score.

Seamless Integration and Conformity with TMIs

This criterion is used to indicate if the UA will be able to seamlessly integrate and conform to any TMIs. If checked this criteria provides a full score.

Flight Plan Adherence to Sector Approval and MAP Evaluation

This criterion is used to indicate that the flight plan will be evaluated against MAP values in each sector during the approval process. If checked this criteria provides a full score.

UAS Location and Navigation

This criterion is used to indicate that the UAS has a reportable location and is capable of navigating via its own means. If checked this criteria provides a full score.

Sense and Avoid

This criterion is used to indicate that the UAS is capable of sensing and avoiding traffic. The user may mark this criterion as satisfied, if the use case for the UAS does not apply for this criterion. If checked this criteria provides a full score.

GPS Free Operation

This criterion is used to indicate that the UAS will be able to operate safely without GPS. The criterion is important in determining that the UAS can reach alternate landing sites and diverts using available backup equipment (i.e. if the back-up navigation is via terrestrial navigation using VORs, then the off nominal routes must be able to fly such alternate routes). If checked this criteria provides a full score.

ADSB Free Operation

This criterion is used to indicate that the UAS will be able to operate safely without ADSB. This criterion is additionally satisfied if the use cases for the UAS do not require ADSB or there is sufficient backup reporting mechanisms in place (squawk Mode C altitude, etc.) If checked this criteria provides a full score.

NORAC Safe Operation

This criterion is used to indicate that the UAS will be able to operate safely during NORAC. If checked this criteria provides a full score.

Lost Datalink Operation

This criterion (goal G7.1) is used to indicate that the UAS will be able to safely operate if the control datalink is lost. This criterion rests on the capability of the UAS to function and operation on inertial sensors (or similarly functioning hardware). This criteria is not intended to cover mission ditching or diverting due to component failure, as these are covered by other goals/scoring criteria. If checked this criteria provides a full score.

Flight Plan Contingency Routes

This criterion will check to see if flight plan has contingency routes available along its route. Since at this time, only contingency routes (i.e. routes for emergency conditions such as an engine-out case) are being considered, the algorithm is essentially applying stronger constraints on the goals than might otherwise be needed.

This scoring algorithm starts by examining the first and last "at altitude" points along the flight plan for contingency routes – for the moment it is assumed that the UA will be able to travel to the departure or landing sites in the ascent and descent portions of the flight plan respectively. The "at altitude" portion of the flight plan is then divided into segments of a distance equal to 2/3 of the "at altitude" glide distance. The midpoint of each segment is then checked for available contingency routes.

After the flight plan has been subdivided and contingency routes have been found (or not found) for each segment, the score received for the entire flight plan will be:

$$S_{Total} = \frac{S_{CR}}{L_{Path}} \cdot \sum_i L_i \cdot \begin{Bmatrix} 0, \mathit{no\ contingency\ routes} \\ 1 - \prod_{j=1}^{N}\left(1 - \frac{f_j}{100}\right), \mathit{for}\ 1 - N\ \mathit{contingency\ routes} \end{Bmatrix} \tag{4}$$

S_{Total} = the total score for this criteria.

S_{CR} = the Contingency Route criteria score from Table 2.

L_{Path} = the total length of the "at altitude" portion of the flight plan.

L_i = the length of the segment / subdivision portion of the flight plan.

f_j = the contingency route score (from 0 to 100) for contingency route *j*.

Scoring Normalization

As mentioned in the beginning of the previous section, the Flight Path Scoring algorithm is a normalized (to 100) function along the filed mission flight path using the GSN *goals/criteria* as the function variables. The total score for any given flight plan is simply the sum of all the criteria scores scaled by a normalizing factor.

$$S = N_F \cdot \sum_i s_i \tag{5}$$

S is the flight path score, s_i = the criteria score for the i^{th} criteria, and N_F = the normalization factor, with:

$$N_F = \frac{100}{\sum_i M_i} \tag{6}$$

M_i is the maximum criteria score for the i^{th} criteria.

In addition to each flight plan having a reported score, S, each segment of the flight plan, determined by the segmentation/subdivision from the Contingency Route criteria process and intersections with airspace elements, will be given an individual score representing the individual contributions of all the criteria scores for that segment. Segments along the flight plan can then be color coded according their individual scores. For example, green for the maximum possible segment score, red for a 0 to low score, and yellow for a score that falls between the two outer ranges. Figure 13 shows an example of a segmented flight path with colored scores for each segment.

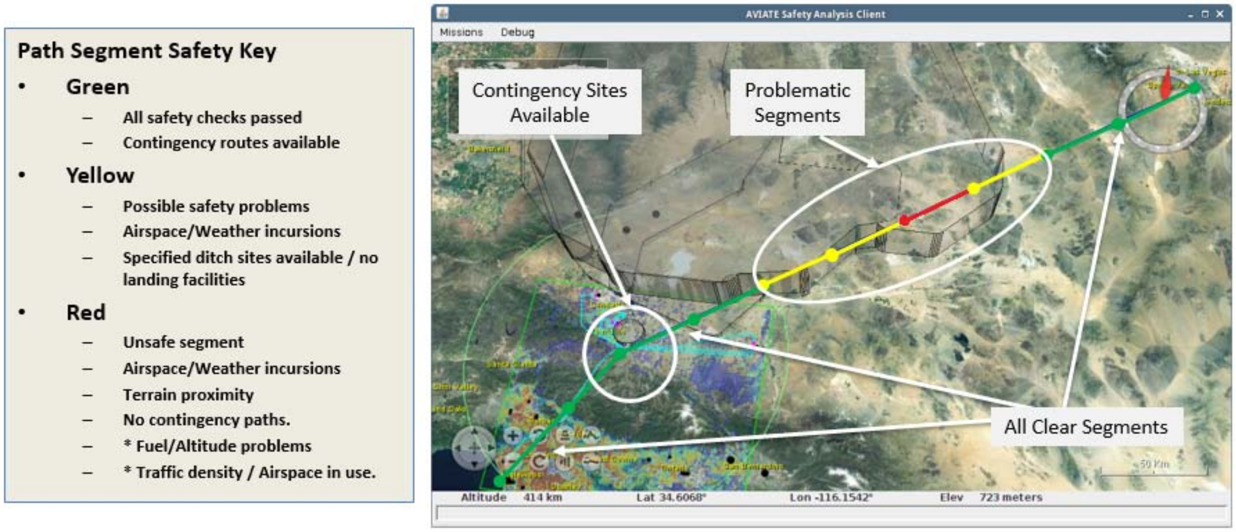

Figure 13. Flight Path Scoring by Segment

Future Development

Future developments of the Aviate system may come via extensions of one of the two existing clients, or the need to create additional clients. Development of clients for two of the use cases mentioned earlier, the FAA Flight Planning and Approval Tool (Use Case 2) and the UAS Pilot Contingency Advisor Tool (Use Case 5) would be straight forward and follow a similar development path as the two existing clients.

The last use case, the On-board Contingency Advisor tool (Use Case 6) would require a longer planning and development process in order to translate the algorithms into the format and size needed to fit into the on-board UA systems.

Furthermore, additional development could arise from the need to integrate some or all of the algorithms present in the Aviate toolbox into an existing system, providing routines for path generation, path scoring, and/or data extraction and use.

Summary

The Aviate system provides a set of tools, analyses, and algorithms creating a methodology for providing safety scores of UA flight plans and generating and scoring contingency routes. It performs multiple rapid analyses of data:

- Identifies divert locations, based on planned route, prior to launch or in flight
- Safety assesses plans and optimizes divert routes around hazards (WX, NOTAM, ATC, etc.) and validates new routes
- Plans and scores routes between points (including divert locations) based on safety considerations

Additionally, the Aviate system provides the following benefits:

- Route safety analysis based on “hazard avoidance” - Population, Weather, Airspace, Traffic
- Metric-driven analyses and optimization of contingency routing - Dynamic safest route selection
- Advisory tool for ATC and UAS pilots - Actionable information for best contingency decisions
- Support of multiple client types - Safety Analysis, ATC Support, PIC Support, Onboard UAV

References

[1] RTCA DO-344 (2 Volumes), 2013, Operation and Functional Requirements and Safety Objectives, RTCA Inc.

[2] Spriggs, John, 2012, GSN – The Goal Structuring Notation, Springer Publishers.

[3] Kelly, Tim and Weaver, Rob, 2004, The Goal Structuring Notation – A Safety Argument Approach, Department of Computer Science and Department of Management Studies, University of York, York.

[4] Dechter, R., Judea, P., 1985, Generalized best-first search strategies and the optimality of A*, Journal of the ACM 32 (3), pp. 505–536.

[5] Dolgov, D., Thrun, S., Montemerlo, M., Diebel, J., 2008, Path Planning for Autonomous Driving in Unknown Environments, In Proceedings of the Eleventh International Symposium on Experimental Robotics (ISER-08).

[6] LaValle, Stephen M., 2006, Planning Algorithms, Cambridge Press, sec. 15.3.1, http://planning.cs.uiuc.edu/node821.html

[7] Russel & Norvig, 2002, Artificial Intelligence: A Modern Approach 2ed, Prentice Hall, chapter 3.

Acknowledgements

The Aviate Project is a Small Business Innovation Research (SBIR) effort that is supported by the NASA Armstrong Flight Research Center (Edwards, CA).

Disclaimer

The text and graphics within this paper express only the views and/or opinions of the authors as individuals and do not represent endorsements, commitments or polices of their firm, agency or organization.

Email Addresses

Chris Wargo: cwargo@mosaicatm.com

John DiFelici: jdifelici@mosaicatm.com

2016 Integrated Communications Navigation and Surveillance (ICNS) Conference

April 19-21, 2016