DEVELOPMENT OF AN AUTONOMOUS AVOIDANCE ALGORITHM FOR UAVS IN GENERAL AIRSPACE

J. van Tooren*, M. Heni**, A. Knoll*, J. Beck*

* EADS Defence & Security, Military Air Systems, D-81663 München, Germany
** ATENA Engineering GmbH (on behalf of EADS), Dachauer Straße 655, D-80995 München, Germany

OVERVIEW

An autonomous avoidance system for the use in UAVs in controlled and uncontrolled airspace has to be able to ensure that a UAV is able to fulfil the 'Sense-and-Avoid' requirement with an equivalent level of safety to inhabited aircraft in which the pilot is able to monitor the direct surroundings of the aircraft and has direct control over the aircraft systems. The focus of this paper is on the avoidance functionality of the system. For this functionality a decision making architecture is introduced which is able to determine whether an intervention is required and then define an avoidance strategy and generate an avoidance manoeuvre. A modular approach ensures that only those parts of the system which are critical to the safety of the aircraft need to be certified as such. Depending on the scenario the 'Sense-and-Avoid' requirement is fulfilled using a hybrid approach of a reactive and/or deliberate avoidance manoeuvre generation algorithm. The deliberate part ensures sufficient separation between aircraft such that a potential conflict is avoided in the first place. The reactive function acts as a safety net to avoid an imminent collision hazard. Due to the different requirements for each function two types of algorithms are used. The separation function is based on a deliberate short-term path search. The safety net uses a direct command reactive algorithm. Both algorithms are able to handle constraints with respect to airspace, aircraft performance and multiple intruding aircraft.

1. INTRODUCTION

One of the main responsibilities of a pilot in command is to adhere to the 'Sense-and-Avoid' (S&A) requirement, as stated in the Rules-of-the-Air in ICAO Annex 2 [1]. Here it is stated that 'An aircraft shall not be operated in such proximity to other aircraft as to create a collision hazard'.

A pilot adheres to the Sense-and-Avoid requirement by scanning the surroundings of the aircraft, that is, maintaining situational awareness. If a possible conflict is observed, the pilot may make trajectory adjustments, depending on the type of airspace and ATC clearance. If immediate action is required to avoid a collision, the pilot will have to implement an avoidance manoeuvre according to the Rules-of-the-Air or an advisory from an Airborne Collision Avoidance System (ACAS) in such a way that this does not induce another collision hazard. Once clear of the conflict, the pilot will have to plan and implement the return to the original flight plan to continue the flight.

In a UAV, the pilot is not onboard the air vehicle, but communicates with the air vehicle over a data link. In cases where this is not possible, for instance when the data link has been lost, the uninhabited system has to be able to take over the responsibilities of the pilot. This requires that a certain level of autonomy is onboard the UAV (Figure 1). When the uninhabited system takes over the S&A task, it has to guarantee a level of safety at least equivalent to a human pilot onboard the aircraft [2].

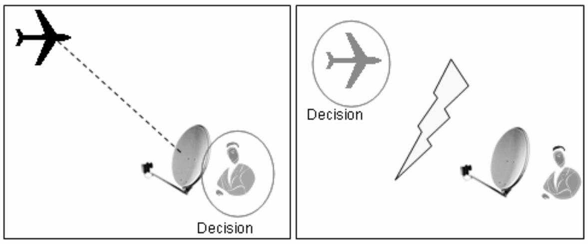

Figure 1: *Conflict avoidance decision making. With established data link the decision is made by the pilot in command. In case of a loss of the data link, the UAV has to make the decision.*

A UAV has to 'behave' comparable to an inhabited aircraft, so that it remains predictable for other users of the airspace. In particular, the trajectories flown by a UAV should not be erratic, but instead be similar to those flown by a human pilot. Furthermore, the UAV should adhere to the Rules-of-the-Air or an advisory from an ACAS system with both cooperative and non-cooperative aircraft in conflict situations.

The first part of this paper presents an overview of the 'Sense-and-Avoid' function and how this knowledge can be used to develop an autonomous conflict avoidance system. The system is based upon a decision making architecture, which has to be able to make two major decisions: First, if and in what time frame, does an intruder aircraft pose a potential collision risk. Second, it must decide upon an avoidance strategy based on the expected time frame and severity of the collision risk.

The second part of this paper discusses how the avoidance strategy is translated into concrete actions of the UAV. An avoidance manoeuvre can either be reactive, deliberate (planned) or a mix of both, depending on the available time and severity of the threat. An algorithm is proposed which is able to generate a manoeuvre which guarantees minimum collision risk subject to mission, airspace, regulatory and aircraft performance constraints.

Finally, in section 9, two example conflict scenarios are discussed and the results of a computer simulation incorporating the proposed system are shown.

2. AUTONOMOUS CONFLICT AVOIDANCE

The avoidance of a collision hazard is managed at two functional levels [3]. The first level ensures sufficient separation between aircraft. The responsibility of this function lies with Air Traffic Control (ATC) or the pilot, depending on the flight regime of the aircraft (IFR, VFR) and the class of airspace the aircraft is flying in. The second functional level is the avoidance of imminent collisions, in the event of a failure of the separation function. Together, the two functional levels enable an aircraft to adhere to the 'Sense-and-Avoid' requirement.

To aid pilots in the Sense-and-Avoid capability, a Traffic Alert and Collision Avoidance System (TCAS) is commonly used to both detect potential conflicts and advise the pilot of a collision avoidance manoeuvre. The TCAS system monitors the traffic in an airspace around the aircraft by listening to the transponder information transmitted by cooperative aircraft. To pinpoint the location of an aircraft, the range and bearing are calculated from the transponder signals and when available the altitude is taken from the transponder message.

When an aircraft is determined to be a threat, the TCAS system will negotiate an avoidance strategy with the cooperating transponder. The resulting strategy is then given to the pilot in the form of a 'Resolution Advisory'. It is then up to the pilot to decide whether to follow the advisory and implement an avoidance manoeuvre that is consistent with the avoidance strategy.

The decision of the pilot to adhere to and implement the advisory is critical, not the advisory itself. In principle, the pilot may deviate from the advisory, if he has information which supports an alternate avoidance strategy, for instance from ATC, or non-cooperative aircraft which TCAS may not be able to 'see'.

In this paper, an autonomous conflict avoidance system is proposed which is modelled analogue to the conventional conflict avoidance process. Analysis of this process identifies the following five main steps, irrespective of the functional level of the collision avoidance system:

1. Sensor consolidation and risk classification
2. System moding and avoidance triggering
3. Avoidance strategy determination
4. Avoidance manoeuvre generation
5. Avoidance manoeuvre implementation

A TCAS system assists the pilot in some of these steps, but in principle the pilot onboard an aircraft has the responsibility for the correct implementation of each step. For UAVs however, all of the steps have to be carried out by the autonomous system if communication between the UAV and the pilot is not possible.

To guarantee an equivalent level of safety for UAVs, it is not sufficient to directly root a TCAS Resolution Advisory to a high level autopilot mode. Information from other sensors that can detect non-cooperative aircraft is required to be able to make an autonomous decision on the avoidance strategy, that is, the information from TCAS needs to be augmented by these sensors. Altogether the autonomous conflict avoidance system must have at least an equivalent amount of information as its human counterpart does.

Furthermore, once the aircraft has manoeuvred away from the conflict, a planned trajectory back to the original flight plan is required. This trajectory should be checked for potential conflicts, which is of particular importance in manoeuvres such as slow overtaking.

Since UAVs have to behave analogously to inhabited aircraft, it is not sufficient to rely on a reactive algorithm only, since such an algorithm is designed to be a safety net. With such a 'TCAS only' system, the UAV will cause constant conflicts and triggers a high amount of Resolution Advisories. Therefore, if the time constraints of the current situation allow to make a short-term re-plan, resulting in a manoeuvre which does not cause a conflict in the first place, this opportunity should be taken full advantage of. A deliberate short-term planning tool is therefore an essential part of a conflict avoidance system to fulfil the first functional level, as described at the start of this section.

3. DECISION MAKING ARCHITECTURE

Since the UAV has to take over some responsibilities from the pilot, a certain level of autonomy is required onboard. Conflict avoidance is one example of such a responsibility and therefore designing such a system requires to determine what type of decision making is necessary. Clough [4] states that it is not the complexity of the problem that dictates the solution, but the task characteristics, since that is what must be accomplished. The following characteristics can be identified and are shown along with the respective choices:

Planning requirements – reactive or deliberate: The conflict avoidance system needs *both a reactive and a deliberative* part. The deliberative algorithm will ensure a conflict free operation of the UAV and the reactive algorithm will function as a flight critical safety net.

Information requirements – local or global: Information required for conflict avoidance has to be gathered *locally on board the UAV*, since the system has to be able to continue to function even without data link. However, it is possible to augment this information using external systems.

Location of processing - central or distributed: Since the system has to be able to continue to function without data link the *location of processing should also be local*, that is, on board the UAV and can not be distributed with external agents.

Rule construction – functional or semantic: A functional system is usually coded using arithmetic (algorithms). Functional systems need clear system boundaries and requirements to function correctly. Ill-defined problems can best be described via semantic expressions, which can be words or symbols. The reactive part of the conflict avoidance system is flight critical. Therefore a *functional rule construction* is used. The deliberate part can in principle be both, but for system transparency and predictability, also *a functional rule construction* is used.

The variety of functional aspects of the conflict avoidance system demonstrates the need for an open decision making architecture with a modular approach which enables the decomposition of the functionalities into small interconnected and manageable parts.

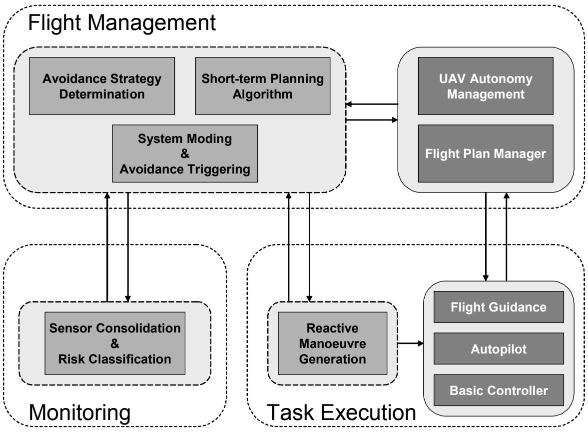

Figure 2: *Decision making architecture. The three main components Flight Management, Monitoring and Task Execution are shown. The conflict avoidance architecture is included in the dashed light gray boxes in all three main modules.*

A conflict avoidance system *monitors* the flight path of the UAV with respect to its surroundings, makes a *decision* whether intervention is required and *executes* an avoidance manoeuvre. These requirements map to a decision making architecture, as shown in Figure 2, which has three main modules with the following functionalities:

- Flight Management
- Monitoring
- Task Execution

The *Flight Management* module contains the mission plan execution, flight plan management and decision logic modules and communicates with both the Monitoring and the Task Execution modules. The *Monitoring* module contains systems which consolidate information from the sensors and monitor the operation of the UAV system. The *Task Execution* module contains all executing functionalities of the vehicle and mission systems, such as flight guidance, autopilot and basic controller modules.

Figure 2 shows that the conflict avoidance system can not be located in a single box within this architecture, but is distributed over the three functional modules and interfaces with various subsystems of the UAV.

The sensor consolidation and risk classification subsystem is located in the Monitoring module. It basically translates the sensor signals into a format which can be used to base a decision on. It sends the situation report to the Flight Management module which will then decide if an intervention is required.

All system moding and avoidance triggering, strategy determination and short term re-planning planning modules of the conflict avoidance system are located in the Flight Management module. Depending on the type of intervention required, deliberate or reactive, a trigger is sent to the corresponding system within the Flight Management or the Task Execution modules respectively, along with the determined avoidance strategy.

If a deliberate re-plan has been filed to the Flight Plan Manager, the Task Execution module will simply carry out the avoidance Flight Plan, as it would do with any other Flight Plan. However, when a reactive manoeuvre is required, the Reactive Algorithm module within the Task Execution module is activated which can directly communicate with the autopilot, overriding the commands from the Flight Guidance computer. Such a direct path ensures that the reactive avoidance manoeuvre is not routed through the Flight Plan Manager.

Not all parts of the complete conflict avoidance system need to be certified as flight critical. Only the part that functions as the safety net would be certified as such. The part of the conflict avoidance system that functions as the safety net comprises of the sensor consolidation and risk classification, the system moding and avoidance triggering, the strategy determination and the reactive algorithm modules.

The part of the conflict avoidance system that ensures that a conflict doesn't occur in the first place does not have to be certified as flight critical, which allows different types of short term re-planning algorithms to be used. The result of the short term re-plan is a flight plan which is filed to the Flight Plan Manager which makes the necessary checks. If the flight plan is not accepted, this does not cause an immediate risk to the safety of the vehicle, since the safety net would still function.

Note however, that this does not mean the short-term re-planning algorithm is not an essential part of the system, since without it, the UAV would continuously cause conflicts resulting in frequent resolution advisories in surrounding aircraft.

4. SENSOR CONSOLIDATION AND RISK CLASSIFICATION

The main output of the sensor consolidation and risk classification function is a report of the traffic situation which can be used by the System Moding and Avoidance Triggering function to base its triggering decision on. Its input consists of sensor data from various sources, which have been pre-processed to various degrees, based on the type of sensor. If available, information from the TCAS system is used as an additional 'sensor'.

The first task is to consolidate and pre-process this information into a single state vector for each intruder aircraft, containing all information required for the risk classification function. This includes the range, range rate, range acceleration, bearing and altitude of each intruder aircraft relative to the own aircraft together with their relative errors.

To quantify the level of risk an intruder aircraft poses, it is not sufficient to only use a distance-based metric which is fixed for each intrusion scenario. This is because the collision risk is dependent on both the predicted distance at closest point of approach and the predicted time it will take to reach this point. The risk is therefore quantified using both a distance-based and a time-based metric.

Zones are defined around an aircraft with their size and shape being defined by a threshold for the minimum allowable distance and the projected time it takes to reach this minimum distance.

The TCAS system uses two zones to classify the risk of each intruder aircraft. The largest zone is used to trigger a 'Traffic Advisory' (TA) to the pilot, which enhances his situational awareness. Inside this zone, a protected zone is defined which triggers the generation of a 'Resolution Advisory' (RA) (Figure 3).

To avoid the UAV causing frequent resolution advisories, the conflict avoidance system should ensure that the UAV will not violate the protected zone of other aircraft in the first place. Therefore, the same zone definitions as used in TCAS are used for the short-term planning algorithm of the conflict avoidance system.

An extra zone is introduced for the reactive part of the conflict avoidance system. This 'Autonomous Avoidance' (AA) zone lies within the protected TCAS zone and is used to define when to trigger a reactive avoidance manoeuvre (Figure 3).

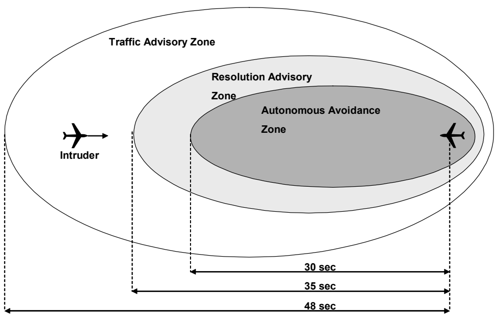

Figure 3: *Risk classification zones: The TCAS Traffic Advisory and Resolution Advisory zones are extended inwards by the Autonomous Avoidance Zone.*

All zones use the distance-based and time-based thresholds as described above. To compensate for traffic density and altimeter errors, the thresholds of the zones are varied, depending on the sensitivity level in which the system operates. This sensitivity level is automatically varied with altitude or can be set by the pilot [5].

Since the zones are dependent on the relative geometry of the intrusion scenario, the zones will alter shape and size, depending on the geometry of the avoidance manoeuvre. This means that the avoidance manoeuvre will have to be flown exactly as planned, both in position and time, since otherwise it can not be guaranteed that the protected zone of the UAV is not violated. This results in a stringent requirement on the path planning algorithm which should only generate flyable flight paths.

5. SYSTEM MODING AND AVOIDANCE TRIGGERING

The System Moding and Avoidance Triggering subsystem (Figure 4) has two functions, both using a rule-based decision making system.

The first function is to determine in which mode the conflict avoidance system should be. These modes include ground, air and intermediate modes which set the constraints on the decision making logic, avoidance strategy and sensor modes.

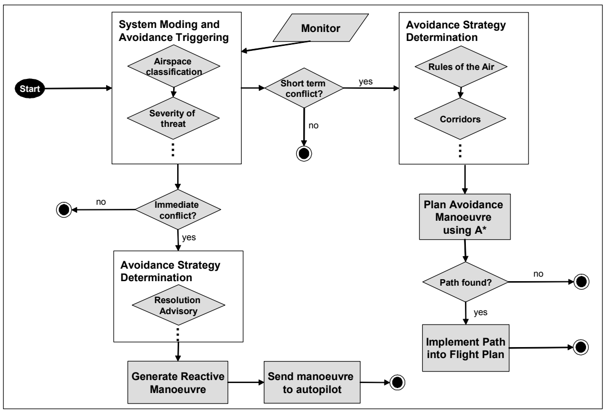

Figure 4: *Avoidance triggering decision flow diagram.*

The second function of the subsystem is to decide when to trigger a deliberate re-plan or a reactive manoeuvre, based on the information from the Monitor module. This decision is based on aspects such as the severity of the threat, the Rules-of-the-Air, the airspace classification and the system mode. The severity of the threat is based on the information obtained from the risk classification function as described in the previous section.

When it is predicted that the protected zone of the UAV will be violated in the near future, the deliberate short-term planning algorithm can be triggered to avoid this from happening. If the Autonomous Avoidance zone is violated the generation of a reactive manoeuvre should be triggered.

The Rules-of-the-Air dictate the levels of the thresholds at which the interventions should take place. For instance, if the UAV has the right-of-way over the intruder aircraft, it should not immediately re-plan, since this would not be consistent with the Rules-of-the-Air. Only if the intruding aircraft does not alter its course, then eventually the UAV should manoeuvre in such a way that a collision hazard is minimized.

Depending on the type of airspace the UAV is flying in, triggering thresholds can be varied. For example, all aircraft or only some aircraft can be in contact or be controlled by ATC. Also depending on the airspace intruder aircraft might be flying under IFR or VFR. Consequently, different algorithm thresholds are required.

Finally, the system mode also has an influence on the triggering thresholds, since it is possible that the system needs to function differently when flying a final approach compared to when it is cruising at its service ceiling.

While flying the avoidance manoeuvres, the system continues to monitor the intruder aircraft to ensure that the assumed trajectory tolerances of both own aircraft and intruder aircraft are not violated. If any of these tolerances are violated, either a re-plan is made or a reactive manoeuvre is initiated, depending on the time available and severity of the risk.

6. AVOIDANCE STRATEGY DETERMINATION

Before an avoidance manoeuvre can be generated, a choice on the avoidance strategy needs to be made. This strategy basically states the constraints under which the generation of the avoidance manoeuvre should be initiated. These constraints result from an analysis of the current threat situation. The detail of the analysis of the threat situation depends on the type of algorithm that is to be initiated.

For a reactive algorithm, all constraints will generally be consolidated into a single advised attitude change. A TCAS resolution advisory is an example of this. Such a constraint type does not allow any flexibility for the generation of the avoidance manoeuvre, which suits a reactive manoeuvre generator.

A deliberative short term re-planning algorithm is able to handle a set of constraints which does not explicitly state the required avoidance manoeuvre. This flexibility of a deliberative algorithm allows more optimal paths to be found within the boundaries set by the constraints.

The details of the strategy are dependent on several factors, analogue to the avoidance triggering thresholds.

First, the Rules-of-the-Air should be adhered to as long as this does not induce an imminent collision hazard. An example of such a rule would be that an UAV which does not have right-of-way avoids the collision hazard by changing its heading to the right. A possible translation into a constraint is to block the airspace to the left of the aircraft track.

If a TCAS Resolution Advisory is received, this can be used as a constraint by stating that the avoidance strategy should not contradict the RA. For instance, when a 'Climb, Climb' RA is issued, the UAV should not be allowed to choose a descending flight path angle. However, lateral manoeuvres are principally still within the solution space.

Airspace and clearance constraints can be defined by using corridors and/or restricted zones.

7. REACTIVE MANOEUVRE GENERATION

The reactive manoeuvre generation algorithm (Figure 5) serves as a safety net, not as an optimal path generator. This means that an acceptable manoeuvre is required, not an optimal manoeuvre. An acceptable manoeuvre is any manoeuvre which ensures the collision hazard is avoided, without inducing a new collision hazard with another aircraft.

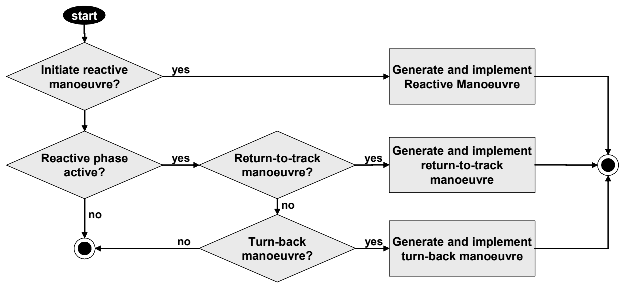

Figure 5: *Decision logic of the reactive manoeuvre phase.*

The reactive manoeuvre algorithm basically consists of the following two phases (Figure 6):

- Reactive avoidance phase
- Guidance to original track phase

The reactive avoidance phase follows the advised track change as stated in the avoidance strategy and determines the necessary commands to the autopilot. The second phase incorporates the guidance of the UAV back to the original track, once the collision hazard has been avoided.

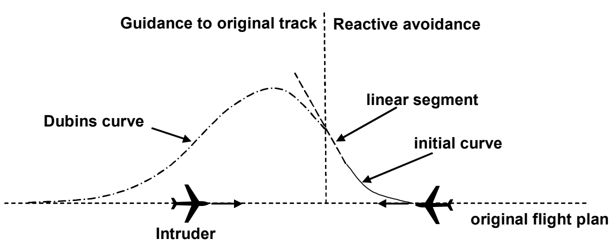

Figure 6: *Reactive manoeuvre generation. Right: reactive avoidance manoeuvre, left: guidance back to original track.*

The standard reactive avoidance trajectory consists of a curve segment, followed by a linear segment. The required orientation (longitudinal, lateral, 3D) and angle change of the curve is provided in the avoidance strategy and the length of the linear segment is dependent on the look-ahead time of the algorithm determining the avoidance strategy.

As the UAV progresses along the reactive avoidance trajectory, the possibilities of implementing either a complete return-to-track trajectory or a turn-back manoeuvre are evaluated to check if these cause a repeated collision hazard. The return-to-track manoeuvre is modelled using a 3D Dubins Curve-Straight-Curve trajectory [6] and the turn-back manoeuvre is modelled using a curve and a linear segment, such that the UAV will fly parallel to the original track.

The second phase is particularly important for manoeuvres where the UAV is slowly overtaking another aircraft: simply flying back to the original track without ensuring the track is free of collision hazards, potentially causes repeated collision hazards. To accommodate the possibility that the UAV will have to fly parallel to the overtaken aircraft before a complete return trajectory is possible, a turn-back manoeuvre parallel to the original track has to be considered as well. Figure 5 shows the decision flow diagram of the different phases of the algorithm.

The avoidance manoeuvre is commanded directly to the autopilot for two reasons. First, the complete avoidance manoeuvre, including return-to-track, is not known at the moment that the reactive part is initiated. The guidance to the original track phase is generated as the reactive part of the manoeuvre is being flown. It is therefore not possible to file the complete avoidance manoeuvre to the Flight Plan Manager as in the case of a fully planned conflict avoidance manoeuvre. Second, a command path through the flight plan manager will generate unnecessary delays and certification issues for this critical function of the conflict avoidance system.

8. DELIBERATE MANOEUVRE GENERATION

The deliberate manoeuvre generation algorithm is used to plan a trajectory around the intruder aircraft in a way to not violate the TCAS RA zone or any additional corridors. The newly generated flight plan should be similar to what a human pilot would do to keep the UAV predictable to other aircraft.

The path search is implemented using an A* algorithm which finds an optimum path around the obstacles from the current location of the UAV to a location back on the original flight plan. The proposed algorithm is able to calculate such a path in a time frame suitable for onboard implementation.

8.1. Short Term Re-planning Decision Logic

The re-planning algorithm can in principle calculate full 3D trajectories. However, arbitrary free 3D trajectories are not necessarily a good solution for an avoidance manoeuvre, in particular in civil airspace, as they create unpredictable and strange manoeuvres especially for other human pilots. Therefore, a multi-step approach is performed to find a possible path where each step describes a complete search space suitable for the current scenario. For each step one resulting trajectory is calculated. The first steps usually start with reduced geometry, for example searching a 2D solution only. If the result complies with all constraints and yields a suitable trajectory the algorithm stops and the resulting trajectory can be integrated into the flight plan. This is repeated until no more suitable search scenarios exist or the maximum calculation time is exceeded (Figure 7).

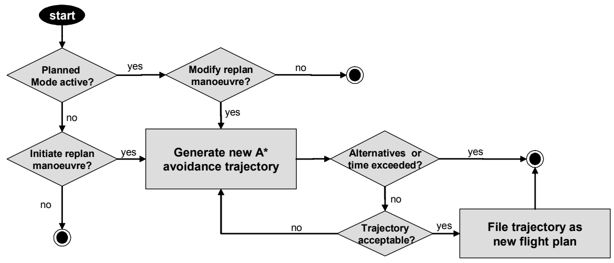

Figure 7: *Decision logic of the deliberate manoeuvre phase.*

If no path was found in any of the searches, nothing can be and nothing is done. In the next monitoring cycle the conflict alert is triggered again and the process repeats with the updated scenario. This procedure continues until either a path is found or the reactive safety net algorithm is triggered, which will then allow to penetrate the TCAS zones. Note however, that it is highly unlikely that no path is found as the 3D space around an aircraft has plenty of possible paths, even in crowded airspace.

The following list shows the avoidance strategy for a typical search scenario (without additional ATC information) listed in order of preference:

1. A 2D (lateral) path search evading the obstacle to the right.
2. A 2D (longitudinal) path search evading the obstacle above or beneath.
3. A full 3D path search evading the obstacle to the right above or beneath
4. A 2D (lateral) path search evading the obstacle to the left
5. A full 3D path search evading the obstacle to the left above or beneath

8.2. The A* Algorithm

A* is a tree search algorithm that finds a path from a given initial node to a given goal node [7][8]. The algorithm guarantees to always find a solution if one exists. A* is a best-first search algorithm, that is, it expands more promising nodes first. Which nodes are promising is determined using a heuristic estimation function H which estimates the cost from the current node to the goal. Including a heuristic estimation function in the algorithm can result in significantly better performance than pure breadth-first or depth-first algorithms. The amount of nodes expanded can depend drastically (between polynomial and exponential in the length of the solution) depending on the quality of the heuristic estimate.

Each expanded node is assigned a cost G, which is the current path cost from the start node to the current node, and the cost H, which is the estimated cost from the current node to the goal. The overall cost for each node is therefore F = G + H. All expanded nodes are processed in order of this overall cost F. An example of the A* node expansion is shown in Figure 8.

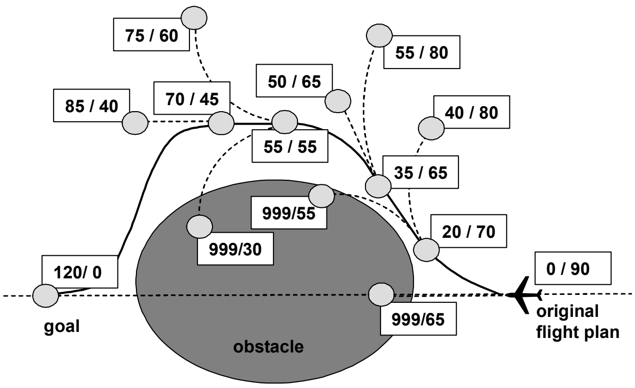

Figure 8: *A* path search around an obstacle (gray). Shown are some expanded nodes resulting from typical motion primitives with their [G / H] cost noted in the attached boxes. Note, that for simplicity not all possible segments are shown. The least cost path is drawn solid.*

A* guarantees to find an optimum path, that is a path with least cost between start and goal node, if the heuristic function underestimates the real cost. However, larger estimates usually yield less expanded nodes. Thus, if no optimum path is required, it can be beneficial to overestimate the cost to the goal. The paths found will then still be good but not necessarily optimum paths. This trade-off between speed and accuracy can be used to tune the algorithm.

The main difficulties with an A* path search is the memory and time consumption of the algorithm, especially if it needs to expand a lot of nodes. Depending on the choice of the heuristic function and the motion primitives used, the amount of nodes can rise tremendously. We use a speed efficient A* implementation where the open and closed node list inherent to A* are implemented as a combination of lists and hash tables for fast lookup. Node insertion is done via binary search on the open list.

8.3. Cost and heuristic estimate functions

The cost function G determines what path is considered as the optimum one. Optimization can be by any criterion, but is usually the length of the resulting path. We use the path length as cost function but additionally include contributions of possible violations of buffer zones around the intruder aircraft, that is, paths too close to the intruder are considered to be more expensive. Furthermore, we add a contribution judging the 'niceness' of the resulting path such that, for example, high load factors along the path are avoided.

The heuristic estimate H uses the distance from the current node to the goal node along a 3D Dubins trajectory [6] connecting both nodes. This is slightly larger than the pure Euclidean distance between the two points but results in a more accurate but still underestimating function H.

8.4. Motion Primitives

A* is often implemented as search over a grid space. However, this is not necessary for A* and in the case of conflict avoidance for aircraft it is actually more convenient to implement it as a search over nodes of motion primitives, that is short trajectory segments [9]. Each of these motion primitives is chosen such that it generates a flyable, smooth trajectory valid for the current state and the performance limits of the aircraft.

The exact form and length of these segments are determined by a manoeuvre space generator which generates a set of possible motion primitives for each current node of the algorithm. The set of possible elements comprises of typical flight elements such as linear segments, curve segments but also more complicated elements like (3D) Dubins sets. The latter is a combination of curve and linear segments to smoothly combine any two points in space in such a fashion that a trajectory of minimal length with given average curvature and tangents at the start and end points is created. Using these Dubins sets, in particular for the return to the original trajectory, allows to keep the A* search tree much smaller compared to building it by only linear and curve segments. Table 1 depicts a set of possible manoeuvres resulting from the manoeuvre space generator.

After the generation of all possible manoeuvres for the current search node, this set is filtered through a manoeuvre space classifier which checks the segments for consistency with the current flight situation. Elements which would for example violate altitude or aircraft performance limits are dropped at this point. Only motion primitives which pass through this filter are fed into the A* algorithm to build the next set of nodes.

Segment type	**Duration**	**Remark**
Linear	3.7 sec	
Right turn	11.0 sec	30 degree, 0.4 g (***)
Right turn	3.7 sec	10 degree, 0.4 g
Dubins set	102.1 sec	Curve-Straight-Curve
…	…	…

Table 1: *Some typical motion primitives generated by the manoeuvre space generator for a UAV flying with 100 m/s. Some elements (***) are only created for step 1 of the algorithm.*

Additionally, we introduce a special feature into the manoeuvre generation. New nodes are fed in two steps into A*: At first, a finite state machine inside the manoeuvre generator tracks the phase of the avoidance manoeuvre (initial avoidance, ..., fly back to the original track) and delivers only a suitable sub-set of motion primitives for this part of the manoeuvre to A*. Paths generated this way can be slightly longer (higher in cost) but are more similar to what a human pilot would do and thus what the human pilot in the intruder aircraft expects. Only if no path can be found, then the remaining segments are injected into the algorithm. This two-step approach allows a very fast search which will find 'good' paths first without loosing the ability to search complicated paths in difficult scenarios.

8.5. Return to the original flight plan

The algorithm aims to return to a goal point along the original flight plan. To determine this goal point, the original trajectory is investigated for a collision free part of sufficient size before the A* algorithm is started. Once such an area is found, a suitable large goal area on the original flight plan is marked as 'goal'. The algorithm can then find a path back to points inside this collision free area. As points farther ahead on the original trajectory usually mean longer paths the algorithm automatically chooses an early return point.

8.6. Obstacles

A trajectory around the intruder aircraft must not violate any forbidden zones. In the A* algorithm this is handled by assigning high or infinite cost to segments leading through undesired or forbidden zones. There are several possibilities for such zones:

- TCAS zones: The TCAS protected zone is forbidden space.
- TCAS buffer zones: A buffer zone around the protected zone with high cost is introduced which can but should not be entered. This accounts, to a certain extent, for measurement errors or unpredicted manoeuvres of the intruder aircraft.
- Forbidden zones: Some airspace can be marked as forbidden zone which the algorithm will not enter at all.
- Desired zones: The algorithm can plan a route in a given desired airspace only, for example to respect ATC cleared corridors.

The size and shape of the TCAS and TCAS buffer zones depend on the relative geometry of the aircraft states. Each node during path search creates a different scenario and consequently these zones also introduce obstacles which permanently change in shape and size. This requires that for each node all zones and zone violations need to be recalculated, which is a major contribution to the CPU time used by the algorithm.

TCAS zones assume straight, unaccelerated flight and include path deviations in the zone definitions itself [see section 4]. However, to determine a possible TCAS zone violation during path planning, the courses of all intruder aircraft need to be known and propagated into the future. Depending on the information about the intruder aircraft available, these courses are either extrapolated linearly or if the flight plan of the intruder is available, for example via ATC, the known plan is used for the planning algorithm.

To account for small deviations of the intruder aircraft flight plan or errors in its extrapolation, an additional buffer zone around the TCAS RA zone is allocated. The path planning finds a route around the buffer zone and therefore deviations occurring only inside this zone can be safely ignored. Only if the intruder deviates more than the size of the buffer zone, a re-replan is required.

9. SIMULATION RESULTS

To demonstrate the conflict avoidance algorithm, two scenarios are discussed. It is shown which decisions are involved, what the planned trajectory looked like and how the UAV is able to follow the calculated trajectories precisely, such that a conflict is avoided.

The risk classification zones, as described in section 4, are independent and different for each intruder aircraft since they are based on the relative geometry of the aircraft states. To depict which zone belongs to which intruder aircraft in a multiple aircraft scenario, it is advantageous to show the zones relative to the intruder aircraft.

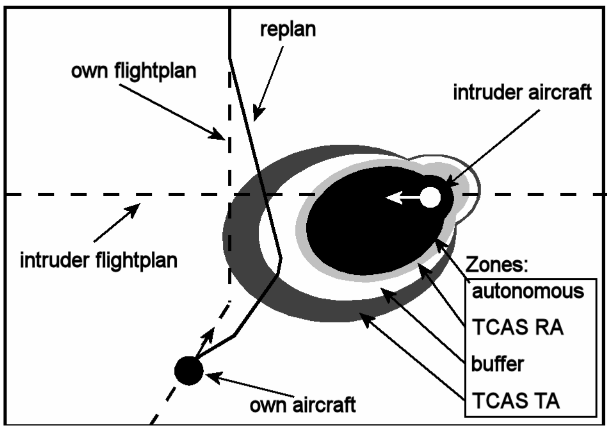

Figure 9: *Legend for the sequential snapshot plots of the example scenarios. Shown are the flight plan of the own and intruder aircraft and the re-planned trajectory. The TCAS trigger zones, the buffer zone and the Autonomous Avoidance zone (see also Figure 3) are shown as well.*

Figure 9 shows a legend of how a scenario will be presented in this paper. It includes the location of the UAV and the intruder aircraft along with their respective flight paths. Furthermore, the size and shape of the various risk classification zones are shown. Note that each picture represents only a snapshot of a highly dynamical situation.

9.1. Scenario One: Aircraft From Right

The first scenario involves one intruder aircraft coming from the right, which therefore has the right-of-way. The UAV has been cleared for a corridor with a maximum lateral track offset of 2 nm and a longitudinal offset of 500ft. The original flight plan of the UAV includes a heading change to the north, which should be taken into account for both the triggering logic and during the re-planning of an avoidance manoeuvre.

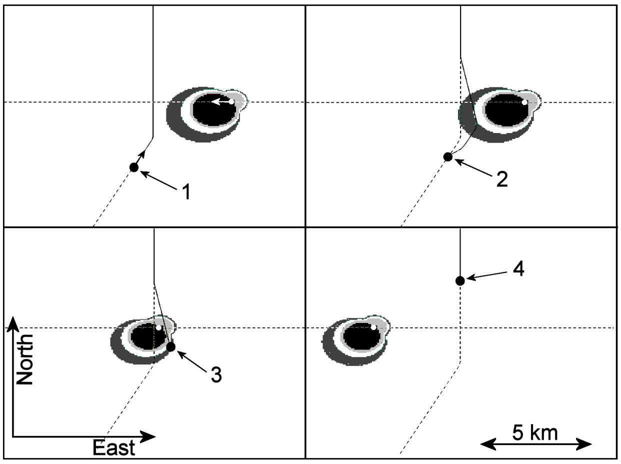

Figure 10: *Snapshots at four different times (top left, top right, bottom left, bottom right) of the simulation. Shown are the trajectories and zones as described in Figure 9.*

In Figure 10 the intrusion scenario is depicted at four points in time, along with the decisions made by the UAV during the encounter with the intruder aircraft. At point 1, the intruder is declared a threat and the short-term planning algorithm initiates a re-plan. At point 2, the new flight plan is implemented and the UAV starts the avoidance trajectory. At point 3, the UAV reaches the closest point of approach during the avoidance manoeuvre, staying clear of the buffer zone around the TCAS protected zone. Finally, the UAV returns to the original flight plan in point 4 and the avoidance manoeuvre is completed.

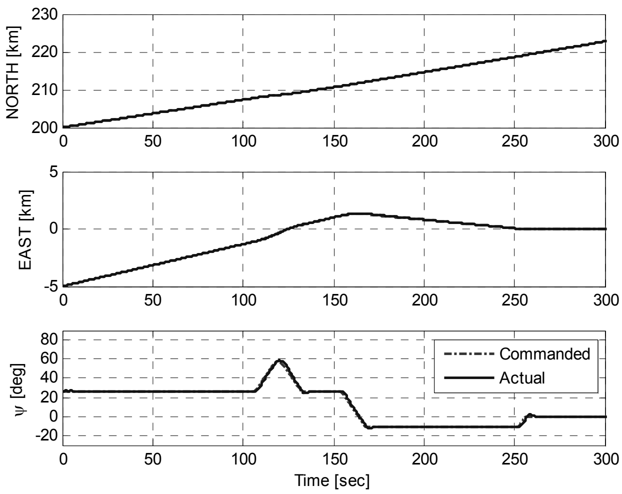

Figure 11: *UAV time histories (trajectory component north, trajectory component east, heading). All plots show both the commanded and actual values which lie directly on top of each other, indicating that the UAV was able to follow the trajectory very precisely.*

It can also be seen in Figure 10 how the zones change shape and size, depending on the trajectory flown by the UAV. This demonstrates the fundamental problem of planning a trajectory where the obstacles are dependent on the trajectory itself.

The simulation used for the examples includes the non-linear dynamics of a typical UAV including an aerodynamic model, basic controller, autopilot and an auto-throttle. The time histories of this scenario, as depicted in Figure 11, show that the UAV was able to fly the planned trajectory accurately.

9.2. Scenario Two: Multiple Aircraft

The second scenario involves a multiple aircraft scenario with an aircraft coming from the right and an aircraft head-on. The aircraft from the right has the right-of-way and the aircraft head-on has to be avoided to the right, according to the Rules-of-the-Air.

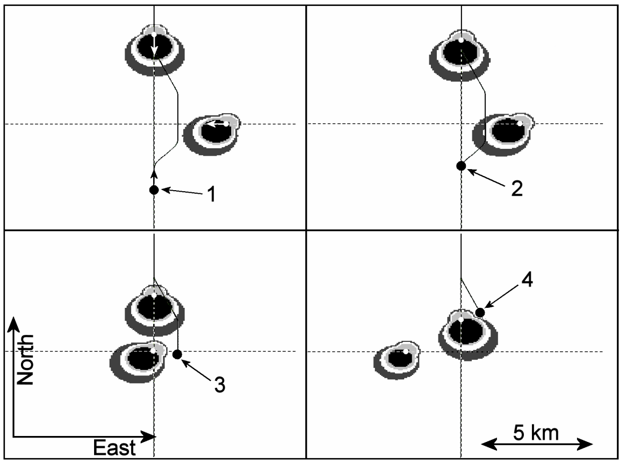

Figure 12: *Snapshots at four different times (top left, top right, bottom left, bottom right) of the simulation. Shown are the trajectories and zones as described in Figure 9.*

Figure 12 shows the scenario as flown by the UAV at four points in time. In the first part of the figure, the intruder from the right is considered a threat and triggers the re-planning algorithm. At point 2, the re-plan is filed to the Flight Plan Manager and the avoidance manoeuvre is started. The trajectory is planned slightly wider than usually necessary for an intruder from the right, because during the same avoidance manoeuvre it is also necessary to avoid violating the protected zone of the head-on aircraft. In the third part of the figure, it can be seen that the return-to-track trajectory is postponed such that no conflict is caused with the head-on intruder. Finally in point 4, the UAV returns to the original track and continues its flight.

The actual range between the UAV and the intruder aircraft coming from the right is shown as a function of time in Figure 13. The figure also shows the range in the case the UAV would have stuck to the original flight plan, that is, having a collision. Due to the avoidance manoeuvre, the range first decreases more rapidly than on the original flight plan. This is because the UAV turns into the direction of the intruder aircraft to be able to pass around behind it. However, the range finally reaches a minimum at a distance larger than the buffer zone around the TCAS zone, therefore correctly avoiding a TCAS zone violation.

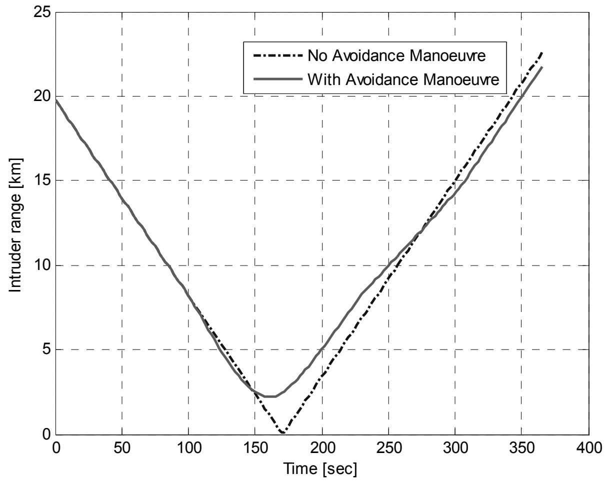

Figure 13: *Actual miss distance for the intruder aircraft with and without the UAV performing an avoidance manoeuvre.*

10. CONCLUSIONS

For UAVs to be allowed to fly in both controlled and uncontrolled airspace, an autonomous avoidance system has to be able to take over the responsibility of the 'Sense-and-Avoid' requirement in events such as a loss of data link.

In the case that the UAV is in controlled airspace, it only needs a last-ditch collision avoidance function to account for unexpected behaviour of the intruder aircraft or failure by ATC to ensure separation. However, an additional level of functionality is required if the UAV is not under such control or encounters intruders inside its cleared flight corridor. Then a separation of the UAV with surrounding traffic should ensure that no collision hazards occur in the first place. These different requirements are built into the two functionalities of the avoidance algorithm.

A decision making architecture has been proposed within this paper which allows the conflict avoidance system to monitor the risk of a collision hazard, decide whether an intervention is required and trigger the necessary avoidance algorithm, either reactive or deliberate. Depending on which algorithm is triggered, a new flight plan is either filed to the Flight Plan Manager or direct commands are sent to the Autopilot. The modular approach ensures that only those parts of the system which are critical to the safety of the aircraft need to be certified as such.

The first functionality requires a reactive algorithm which is critical and acts as a safety net in case the aircraft separation function has failed for any reason. The reactive algorithm is able to handle constraints imposed by the Rules-of-the-Air, resolution advisories from external systems, airspace restrictions and aircraft performance but will not produce optimal trajectories.

The second functionality is a deliberate short-term planning algorithm which is able to handle constraints with respect to airspace and aircraft performance and dynamical constraints formed by zones around the intruder aircraft of which size and shape are dependent on the planned trajectory itself. It finds a suitable trajectory which mimics trajectories flown by human pilots, an important factor to remain predictable to other users of the airspace. The proposed algorithm is shown to fulfil these requirements in a realistic UAV simulation of which some results have been presented in this paper.

11. REFERENCES

[1] ICAO Annex 2, "Rules of the Air", Chapter 3.2, 2006.
[2] UAV Task Force, "A Concept For European Regulations For Civil Unmanned Aerial Vehicles (UAVs)", JAA/EUROCONTROL, pp. 33, 2004.
[3] J. Krozel, T. Mueller, G. Hunter, "Free Flight Conflict Detection and Resolution", AIAA-96-3763, pp. 1-11, 1996.
[4] B.T. Clough, "Relating Autonomy to a Task – Can it be done?", AIAA 2002-3445, pp. 1-9, 2002.
[5] ICAO Annex 10, "Surveillance Radar and Collision Avoidance Systems", ICAO Annex 10 Vol. IV, 2002.
[6] L.E. Dubins, "On Curves of Minimal Length with Constraints on Average Curvature and with Prescribed Initial and Terminal Positions and Tangents", American Journal of Mathematics, Vol. 79, pp 497-516, 1957.
[7] P.E. Hart, N.J. Nilsson, B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", IEEE Transactions on Systems Science and Cybernetics SSC4 (2), pp. 100-107, 1968.
[8] P.E. Hart, N.J. Nilsson, B. Raphael, "Correction to: A Formal Basis for the Heuristic Determination of Minimum Cost Paths", SIGART Newsletter, 37, pp. 28-29, 1972.
[9] N.D. Richards, M. Sharma, D.G. Ward, "A Hybrid A*/Automation Approach to On-line Path Planning with Obstacle Avoidance", AIAA 2004-6229, pp. 1-17, 2004.