THE GEARS CONFLICT RESOLUTION ALGORITHM

Richard Irvine

Eurocontrol Experimental Centre,
BP 15, 91222 Bretigny-sur-Orge, France.

richard.irvine@eurocontrol.fr

Abstract

An algorithm is described which finds a set of flyable, conflict-free trajectories for an aircraft which must fly through an environment containing obstacle aircraft whose trajectories are known. The algorithm makes systematic use of a manoeuvre generator and a conflict detector. The manoeuvre generator may embody a variety of aircraft behaviour models which might, for example, take into account standard turn behaviour and wind predictions. The conflict detector may embody a variety of definitions of what makes a trajectory unacceptable, and might, for example, take into account growing uncertainty in the future positions of aircraft. Correctness of solution trajectories is readily demonstrable. The algorithm may be used to construct trajectories which consist of sequences of manoeuvres of a single type, for example, turns only, climbs and descents only or speed changes only, or sequences of manoeuvres of different types. It may be used for prioritized planning of the motion of a number of aircraft. Results obtained from use of the algorithm in a series of simulations are summarised. The algorithm might also be used in operational applications (ground or air-based) should conflict resolution assistance be required or to solve motion-planning problems in domains other than air traffic control.

1. Introduction

1.1 Problem Statement

In the context of air traffic control the term 'conflict' denotes a loss of required separation between aircraft. For example, in current practice aircraft flying en-route must always be separated horizontally by at least five nautical miles or vertically by at least 1000 or 2000 feet, depending upon their altitude. Potential conflicts may be detected on the basis of trajectory predictions. In operational systems an air traffic controller knows the flight plans of the aircraft in his sector and sees their current positions and velocities on a radar screen allowing him to perform such trajectory prediction and conflict detection. In simulations these functions may be performed automatically.

To ensure that potential conflicts are avoided, in operational systems or in simulations, the trajectories of one or more aircraft must be replanned. When this replanning is triggered by the detection of potential conflicts it is termed 'conflict resolution'. Trajectories may also be replanned in other circumstances, for example, to give more direct routes to aircraft. A new trajectory assigned to an aircraft should be conflict-free within some time-frame. Furthermore, it must be 'flyable' by the aircraft, which is constrained by its own dynamic capabilities.

The primary problem addressed in this paper is that of finding flyable, conflict-free trajectories for a single aircraft (termed the manoeuvred aircraft) which must fly during some time-interval through an environment containing obstacle aircraft whose trajectories are known. The problem is complicated by aircraft dynamics, wind and uncertainty in the predicted positions of aircraft.

This is an 'individual' motion planning problem: the motion of one object is planned in the presence of other moving objects. In 'global' motion planning the motion of many objects is planned. 'Prioritized' planning[1] transforms the global planning problem into a sequence of individual planning problems by defining an order of priority and applying an individual planning technique one object at a time. Given a suitable order of priority, the solution proposed here may be used for prioritized planning.

It is expected that the algorithm will find applications in free-flight and free-route simulations but could also have operational applications (ground-based or airborne) if, due to time or workload pressures, conflict-resolution assistance is required. The result of the resolution process is, in general, not one trajectory, but a set of trajectories. In an automated simulation a solution trajectory can be chosen from this set in accordance with optimisation criteria. If the algorithm were being used to provide resolution assistance, a user would simply select his or her preferred solution. The algorithm may also have applications outside of air traffic control.

1.2 Background

The ARC project[2] investigated the feasibility of automatic conflict resolution in dense airspaces. The

ARC resolution algorithms were based on the concept of forbidden area: a forbidden area is a region which could be reached by a subject aircraft, known as the manoeuvred aircraft, were it to perform any manoeuvre within a range of manoeuvres of a particular kind (*e.g.* heading change, speed change, climb or descent) which would bring it into conflict with one of the obstacle aircraft in its environment. For example, a manoeuvred aircraft might come into conflict with an obstacle aircraft were it to perform a turn to any of a range of headings, see Fig. 1.

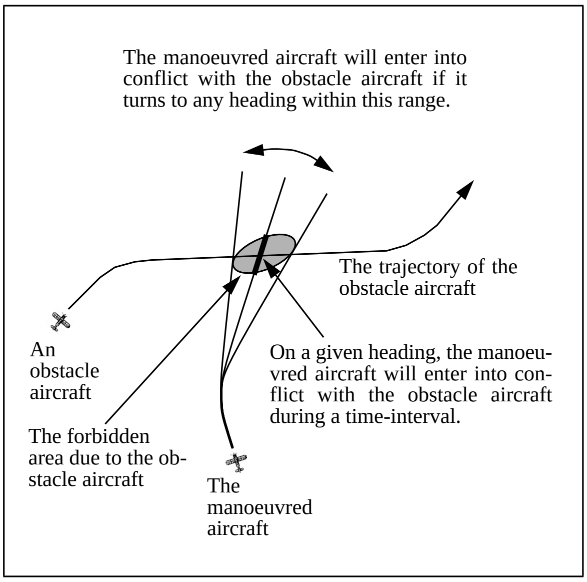

Fig. 1: A Forbidden Area

The position and shape of a forbidden area depend upon the kind of manoeuvre which is being considered, the trajectory of the obstacle aircraft, the definition of conflict (or separation standard) which is being applied and the point and time at which the manoeuvre begins. Assuming a fixed separation standard, an obstacle aircraft flying in a straight line and at the same height as the manoeuvred aircraft gives rise to a forbidden area which is roughly elliptical. However, these forbidden areas may have surprising, concave shapes due to turns of the obstacle aircraft and they may begin and end abruptly due to vertical movements.* Trajectories for the manoeuvred aircraft consisting of manoeuvres which do not cross forbidden areas are necessarily conflict-free.

The approach to lateral conflict resolution (resolution by heading-changes) taken in ARC included the following steps:

1. Identify all forbidden areas (within a range of turn angles)

2. Find (polygonal) approximations which enclose the real forbidden areas.

3. Identify free-space surrounding the forbidden areas. (This is equivalent to detecting and consolidating overlapping areas).

4. Find trajectories which lie entirely within free-space *i.e.* which do not cross forbidden areas.

Given simplified assumptions about aircraft behaviour (*e.g.* they fly at constant speeds, turn instantaneously, there is no wind) and applying a simple separation standard (*e.g.* a conflict occurs when the distance between two aircraft is less than a fixed distance) these steps can be implemented. However, as more realistic assumptions are taken into account (such as turn behaviour and wind predictions) they become increasingly difficult to implement correctly. Finally, this approach breaks down if the separation standard depends upon how the manoeuvred aircraft approaches an obstacle aircraft (*e.g.* to take into account differences in across- and along-track uncertainty or the growth of uncertainty with time) since the approximation of forbidden areas can no longer be separated from consideration of trajectories of the manoeuvred aircraft.[3]

Silbert[4,5] describes a motion planning algorithm, named the Event Step Algorithm, which can be used to find paths avoiding polygons which may be static or moving. This algorithm has two levels of recursion, a lower level which finds possible next ‘steps’ from a starting point and an upper level which constructs a tree of steps. Turning points on solution trajectories for the manoeuvred vehicle are vertices of the (obstacle) polygons. For the static two-dimensional case this algorithm will find the shortest path. In the dynamic case it is assumed that obstacles move with constant speed and direction. It is stated that in this case the problem of finding the optimal path is NP hard. A particularly attractive feature of this algorithm is the way in which it builds a self-consistent set of pertinent obstacles and avoiding trajectories, without having to identify all of the obstacles in the environment with which the manoeuvred vehicle could enter into conflict.

A survey of conflict detection and resolution modelling methods is presented by Kuchar and Yang[6].

1.3 Present Contribution

The algorithm presented here is structurally similar to the Event Step algorithm described by Silbert[4,5], there being two levels of recursion, one which identifies possible next manoeuvres from a turning point and another which pieces these manoeuvres into trajecto-

* The trajectory of an obstacle aircraft can also give rise to more than one forbidden area. This may even occur for an obstacle aircraft flying along a straight line within a certain range of angles of approach, provided its speed is greater than that of the manoeuvred aircraft.

ries. However, obstacles are not modelled as polygons and turning points on solution trajectories are not predefined points (vertices) on the perimeters of obstacles. Instead turning points (or more generally trajectory change points) are chosen to lie on manoeuvres which are, to a close approximation, tangent to forbidden areas. Each manoeuvre (and turning point) corresponds to the ‘visible’ extremity of a forbidden area. These manoeuvres are found by a process which resembles ray-tracing. This approach caters straightforwardly for obstacle aircraft which change direction. Even in the case of obstacles which do not change direction the resulting solution trajectories may be shorter and may have fewer turns than would be the case if the turning points were predefined points on the perimeters of obstacles. Although the idea of forbidden area is essential to understanding the algorithm, no attempt is made to locate forbidden areas, to find approximations which enclose them, or to identify the free-space which surrounds them. In this way the difficulties encountered with these steps in the ARC algorithm are avoided.

No particular aircraft behaviour model is assumed. Rather this is hidden from the algorithm in a restricted form of trajectory predictor, termed a ‘manoeuvre generator’. This allows a high degree of flexibility in the modelling of aircraft behaviour. The manoeuvre generator might, for example, take into account aircraft performance, standard turns and wind predictions. Depending upon the realism of the manoeuvre generator, the resolution algorithm can generate trajectories which are ‘flyable’.

The precise position and shape of a forbidden area depend upon an arbitrary definition of conflict which is hidden from the resolution algorithm within a conflict detector. Once again this allows a high degree of flexibility in the definition of conflict employed. The main purpose of the conflict detector is to take into account uncertainty in the future positions of aircraft. The definition of conflict may depend upon how the manoeuvred aircraft approaches an obstacle aircraft. For example, a fairly realistic conflict detector might assume that the track of each aircraft is known quite accurately, but that the main error is in the predicted time-of-arrival at any point on the track (the along-track error) and that this error grows with time. Another possibility would be for the conflict detector to declare a conflict when the probability of separation loss is found to exceed a threshold. This threshold might itself be time-dependent, for example, having a value close to zero during an initial ‘guaranteed period’, but then increasing with time. (An approach to conflict probability estimation is described by Paielli and Erzberger[7]).

Trajectory predictors and conflict detectors are both components which are well known in air traffic management research.

Finally, and perhaps most importantly, all solution trajectories are necessarily correct, where ‘correctness’ is taken to be judged by the given conflict-detector. As will become apparent from the algorithm description, this is a consequence of the way in which the resolution algorithm makes systematic use of the conflict-detector in order to find solution trajectories.

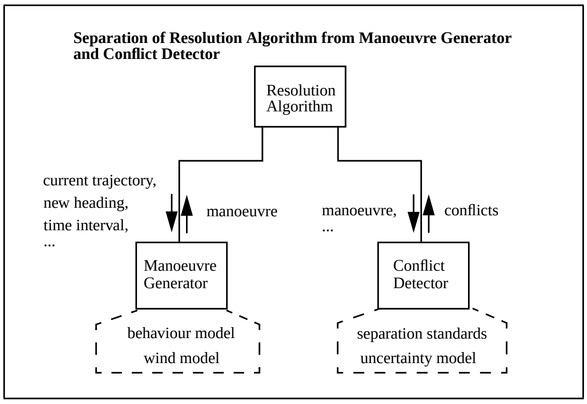

Fig. 2: Separation of Concerns

The algorithm may be used to find solution trajectories in which successive manoeuvres are all of the same kind. In lateral-only resolution the solution trajectories consist of sequences of turns. (It should be pointed out that the manoeuvred aircraft and the obstacle aircraft may, at the same time, be moving vertically). In vertical-only resolution solution trajectories consist of sequences of height changes and in speed-only resolution they consist of sequences of speed changes. The algorithm may also be used for mixed-manoeuvre resolution in which case a solution trajectory may contain turns, height changes and speed changes. It can also be extended to find trajectories which avoid restricted areas in addition to obstacle aircraft.

The algorithm was developed during the course of a project called GEARS (Generic En-route Algorithmic Resolution Service) and consequently it has become known as the GEARS algorithm.

2. Algorithm Description

2.1 Overview

Lateral-only resolution is first described in detail.

From a given starting point and time the algorithm finds a set of conflict-free avoiding manoeuvres which would allow the manoeuvred aircraft to avoid those forbidden areas which are visible from the starting point. The avoiding manoeuvres are found by modifying a preferred manoeuvre in a process which resembles ray-tracing, where each ‘ray’ is a manoeuvre. The avoiding manoeuvres correspond to branches in a tree with the starting point at the root. The end-point of each avoiding manoeuvre is then considered as a new starting point and the process is repeated, so extending the tree. A solution trajectory is a sequence of manoeuvres in the tree from the root to a leaf.

Avoiding manoeuvres can also be generated in the same way but by modifying the vertical profile of the preferred manoeuvre or by modifying its speed profile.

Successive manoeuvres in a solution trajectory may all be obtained by making the same type of modification to preferred manoeuvres. This is termed single manoeuvre type resolution, e.g. lateral-only, vertical-only or speed-only resolution. Alternatively successive manoeuvres in solution trajectories may be obtained by making different types of modification. This is termed mixed manoeuvre resolution.

Forbidden areas are often shown in the diagrams which follow. However, it is important to note that these are shown solely in order to facilitate understanding of the underlying problem: the algorithm does not locate these areas nor approximate their boundaries.

2.2 The Preferred Manoeuvre

From a starting point, the first step in the algorithm is to determine the manoeuvre which the manoeuvred aircraft would prefer to follow during the resolution time-interval in the absence of obstacle aircraft. If the aircraft is already heading towards its destination the preferred manoeuvre will typically be to continue on that heading otherwise the preferred manoeuvre will typically be to turn towards the destination and then continue on that heading until the end of the resolution interval is reached. Considering the vertical profile, the preferred manoeuvre will involve a climb or descent to (or towards) the preferred altitude, if the manoeuvred aircraft is not already at that altitude.

The next step is to perform a conflict check (a '1 against n' check) for the preferred manoeuvre considering all obstacle aircraft in the environment of the manoeuvred aircraft. If the preferred manoeuvre is conflict-free then a solution trajectory has been found. This is the simplest case. In general, however, we expect a number of conflicts to be detected. (These conflicts can be considered to be the intersections of the preferred manoeuvre with forbidden areas due to obstacle aircraft, see Fig. 3).

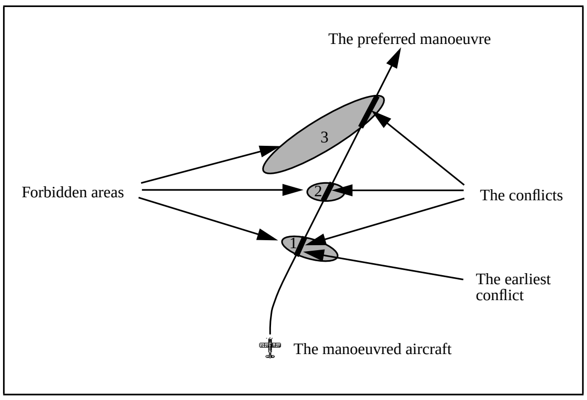

Fig. 3: Conflicts on the Preferred Manoeuvre

When conflicts are found on the preferred manoeuvre, the next step will be to find a set of conflict-free avoiding manoeuvres. However, before explaining how this is done it is helpful to introduce the idea of 'candidate manoeuvres' and to describe how these are found.

2.3 Candidate Manoeuvres

A 'candidate manoeuvre' is a manoeuvre from the starting point which is, to a close approximation, tangent to a forbidden area and which is conflict-free with respect to the obstacle aircraft which gives rise to the area. In other words, a candidate manoeuvre is one which would allow the manoeuvred aircraft to avoid a given obstacle aircraft, but which may not be conflict-free when other obstacle aircraft are considered.

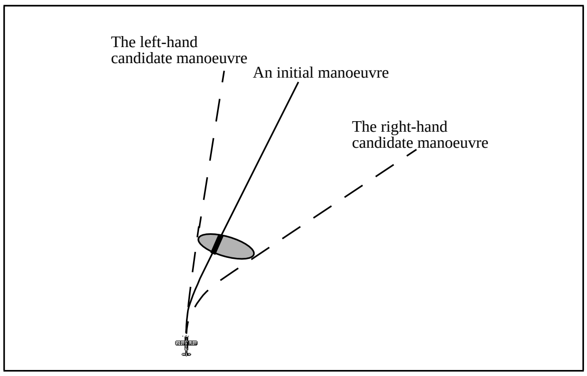

Fig. 4: Candidate Manoeuvres

The next section describes how to find such candidate manoeuvres starting from an initial manoeuvre which would bring the manoeuvred aircraft into conflict with an obstacle aircraft. To begin with, this initial manoeuvre can be thought of as being the preferred manoeuvre from the starting-point. However, it will be seen later in the description that candidate manoeuvres may also be found from initial manoeuvres other than the preferred manoeuvre.

2.4 Searching for a Candidate Manoeuvre

Suppose a conflict with an obstacle aircraft was found on an initial manoeuvre. The most straightforward way of finding a candidate manoeuvre is to take successive steps in heading to one side of the initial manoeuvre. For each heading, a manoeuvre is generated and this is checked for conflict with the first obstacle aircraft (a '1 against 1' conflict check). This process is repeated until a manoeuvre is found which does not conflict with the obstacle aircraft (see Fig. 5).

Apart from the simplicity of this method, it has the advantage that it allows one to build up a picture of the forbidden area, which is useful for visualising the behaviour of the algorithm, or for communicating to a human user which obstacle aircraft a solution trajectory avoids and how it avoids them.

A slightly more refined method would be to take coarse heading steps to find a conflict-free manoeuvre followed by a binary search to find a candidate manoeuvre which is as close to the tangent manoeuvre as required. Other search strategies and optimisations are possible.

Such techniques can be used to find a candidate manoeuvre regardless of the shape of the forbidden area, it may, for example, be concave.

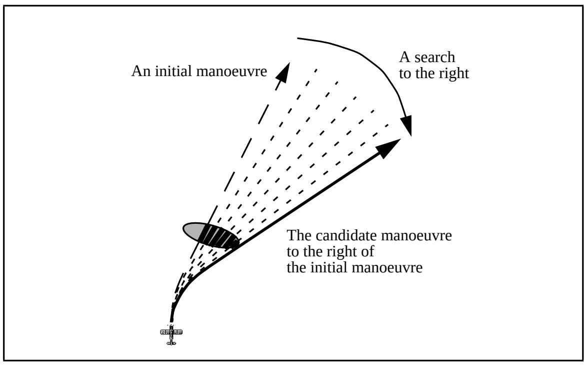

Fig. 5: Finding A Candidate Manoeuvre

Since a possible manoeuvre is considered at every step, the definition of conflict may be both time- and geometry- dependent. Furthermore, since a search for a candidate manoeuvre involves just one obstacle aircraft, only ‘1 against 1’ conflict checks are needed. ‘1 against 1’ conflict checks are much cheaper, computationally, than ‘1 against n’ conflict checks, so that candidate manoeuvres can be found efficiently.

2.5 Finding the Turning Point on a Candidate Manoeuvre

To avoid the forbidden area the manoeuvred aircraft should continue on the candidate manoeuvre at least until a time corresponding to the tangent point. Having begun to avoid the forbidden area, the manoeuvred aircraft may turn again.

If the candidate manoeuvre is found by taking successive heading steps away from an initial manoeuvre, as described in the previous section, an estimate of the tangent time is given by the average of the start and end times of the conflict on the manoeuvre preceding the candidate manoeuvre. If a binary search is used to find the candidate manoeuvre then the tangent time can be found with arbitrary precision.

The tangent time is a lower bound on the time at which the manoeuvred aircraft may turn again. In order to bypass the forbidden area in a single step the manoeuvred aircraft should continue on the candidate manoeuvre at least until a time such that its preferred manoeuvre is conflict-free with respect to the obstacle aircraft giving rise to the area. An intermediate turning time will result in more than one step.

The candidate manoeuvre and turning time together define a turning point.

2.6 Avoiding Manoeuvres

The term ‘avoiding manoeuvre’ is given to a candidate manoeuvre which is conflict-free with respect to all other aircraft between the starting point and the turning point. In other words, an avoiding manoeuvre allows the manoeuvred aircraft to avoid a forbidden area due to an obstacle aircraft and does not create conflicts with any other obstacle aircraft before reaching its next turning point.

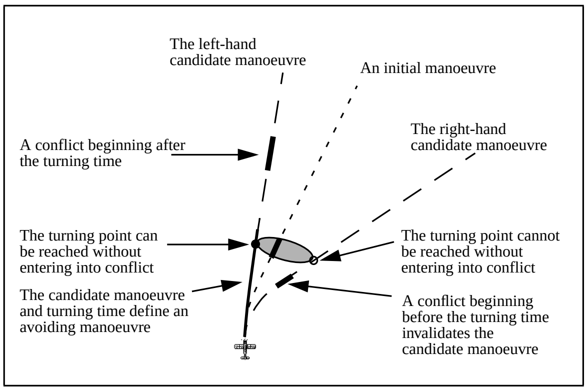

Fig. 6: An Avoiding Manoeuvre

In the example shown in Fig. 6, the manoeuvred aircraft could follow the left-hand candidate manoeuvre until the turning time without entering into conflict. The left-hand candidate manoeuvre and turning time therefore define an avoiding manoeuvre. If the manoeuvred aircraft were to follow the right-hand candidate manoeuvre into would enter into conflict before reaching the turning point. The right-hand candidate manoeuvre and turning time do not define an avoiding manoeuvre.

2.7 Building the Set of Avoiding Manoeuvres

Beginning with the preferred manoeuvre, a conflict check is performed considering all obstacle aircraft. If one or more conflicts are found, the earliest conflict is used to initiate two searches for avoiding manoeuvres: one search will find those avoiding manoeuvres which lie to the left of the preferred manoeuvre, the other will find those avoiding manoeuvres which lie to its right. In other words, the problem of building the complete set of avoiding manoeuvres is split into two sub-problems.

The first step in a search for avoiding manoeuvres is to perform a candidate manoeuvre search in the direction associated with the avoiding manoeuvre search.

When a candidate manoeuvre is found a conflict check (‘1 against n’) is performed throughout the resolution time-interval (*i.e.* up to <u>and beyond</u> the turning time).

If no conflicts are found, or none begin before the turning time, then the candidate manoeuvre is an avoiding manoeuvre and is added to the set of avoiding manoeuvres from the current starting point.

Whenever one or more conflicts are found on a candidate manoeuvre, the earliest conflict (that is the conflict with the earliest start time) is used to restart

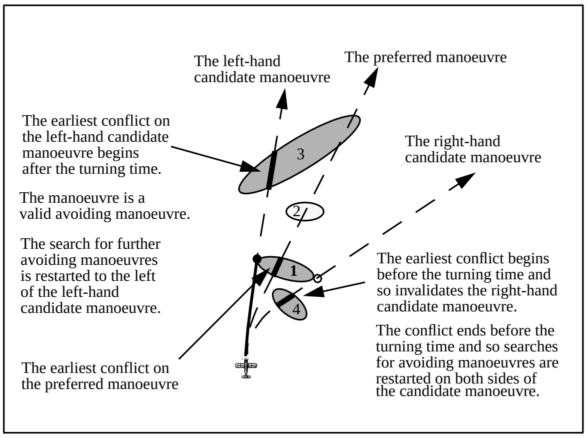

Fig. 7: Building the Set of Avoiding Manoeuvres

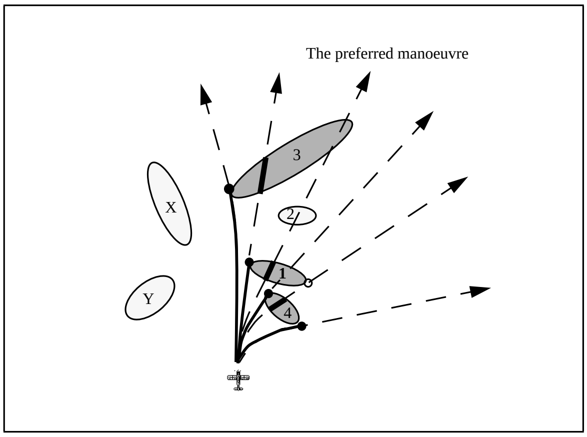

Fig. 8: The Set of Avoiding Manoeuvres

(recursively) one or two further searches for avoiding manoeuvres.

If the earliest conflict ends after the turning time, the search for avoiding manoeuvres is continued in the same direction starting from the candidate manoeuvre. For example, in the case of the conflict with area 3 on the candidate manoeuvre to the left of area 1, the avoiding manoeuvre search will continue to the left of that candidate manoeuvre.

If the earliest conflict ends before the turning time, then there may be avoiding manoeuvres on both sides of the corresponding forbidden area, and so avoiding manoeuvre searches are restarted (recursively) to both sides of the conflict. For example, in the case of the conflict with area 4 on the candidate manoeuvre to the right of area 1, there may be trajectories which pass both to the left and right of area 4 before passing to the right of area 1, and so an avoiding manoeuvre search is initiated to the left of the candidate manoeuvre, but bounded by the turn angle of the preferred manoeuvre, and a second avoiding manoeuvre search is initiated to the right of the candidate manoeuvre, bounded by the maximum turn angle (if one is defined).

As stated above, whenever conflicts are found on a candidate manoeuvre the earliest conflict is used to restart (recursively) one or two further searches for avoiding manoeuvres. In other words, the search for avoiding manoeuvres in a range of turn angles is decomposed into one or two searches in a smaller range of turn angles. However, if the earliest conflict is with an obstacle aircraft which has already been found to block the range of turn angles being searched for avoiding manoeuvres, no further searches are started from this conflict. For example, on checking the candidate manoeuvre to the left of area 4, the earliest conflict found is with area 1. Since the range of turn angles being searched is already known to be blocked by area 1 no further avoiding manoeuvre searches are started from this conflict.

The search for avoiding manoeuvres continues until no further conflicts are found. The complete set of avoiding manoeuvres is shown in Fig. 8.

To simplify the preceding example, the forbidden areas do not overlap, although the same logic is still applicable even if areas do overlap.

It can be seen that the checking of candidate manoeuvres for conflicts throughout the resolution time-interval (and not simply up to the turning time) may lead to the discovery of direct manoeuvres which avoid more than one area. For example, by checking the candidate manoeuvre to the left of area 1 throughout the resolution time-interval the conflict with area 3 is found and the manoeuvre which is then found passing to the left of area 3 also avoids area 1.

It should be noted that no computation is expended exploring the extremities of an area such as area 2, which is completely hidden from the starting point by area 1. Similarly no computation is expended exploring the hidden extremity of an area such as area 3. Also, areas such as X and Y are not investigated in the process of finding avoiding manoeuvres but are also not relevant to the resolution process: assuming the destination of the manoeuvred aircraft is to the top-right of the diagram there is no reason for it to be concerned by these areas. The process of finding avoiding manoeuvres only ‘discovers’ relevant obstacle aircraft, making it both feasible and efficient compared with methods which require all areas to be located before considering avoiding trajectories. Furthermore, this process finds avoiding manoeuvres which avoid overlapping, possibly concave, forbidden areas.

In summary, this sub-algorithm finds a set of avoiding manoeuvres which correspond to the ‘visible’ extremities of those forbidden areas which were found to lie on candidate manoeuvres considered earlier in the search, where ‘visible’, here, means that the turning point can be reached by a single manoeuvre without crossing any forbidden areas. It can be thought of as implementing a kind of machine vision in which the light rays conform to a given aircraft behaviour model and what is seen is determined by the configuration of obstacle aircraft and the given definition of conflict.

2.8 Constructing the Set of Conflict-Free Trajectories

The avoiding manoeuvres are branches in a tree which has the starting point as its root. This tree is extended by repeating the whole process from the turning point of each avoiding manoeuvre. (The number, position and shape of forbidden areas change when viewed from different turning point). A solution trajectory is a sequence of avoiding manoeuvres from the root of the tree to a leaf.

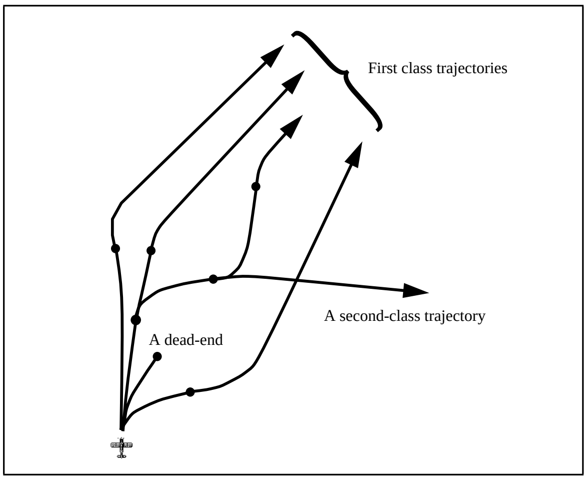

Fig. 9: The Set of Conflict-Free Trajectories

There may be dead-ends in the tree, that is turning points from which no avoiding manoeuvres are found. This can occur if a turning point for the manoeuvred aircraft is so close to one or more forbidden areas that there is insufficient space to turn to avoid it. Also avoiding manoeuvres may be found where the turning-point-time is later than the end of the resolution time-interval. If the manoeuvred aircraft were to follow such a manoeuvre, it would still be avoiding obstacle aircraft at the end of the resolution time-interval and would not be following a preferred manoeuvre. The corresponding trajectory is termed a 'second-class' trajectory. A 'first-class' trajectory is one where at the end of the resolution time-interval the manoeuvred aircraft is following a preferred manoeuvre.

Since every avoiding manoeuvre has been found to be conflict-free, a solution trajectory, which is simply a sequence of avoiding manoeuvres, must also be conflict-free.

The result of the resolution process is, in general, not one trajectory, but a set of trajectories. In an automated simulation a solution trajectory can be chosen from this set in accordance with optimisation criteria. These might include distance from the destination, number of turns, fuel consumption, or some combination of these. If the algorithm were being used to provide resolution assistance a user might simply select his or her preferred solution.

2.9 Rubber-Banding

The algorithm as described so far finds a set of trajectories which is a super-set of the trajectories which are really of interest. Fig. 10 shows two trajectories which will be found by the algorithm. The left-most trajectory includes two successive turns to the right, to pass to the right of areas 1 and 2 and is sub-optimal (at least in terms of the number of manoeuvres required) compared with the right-most trajectory which passes to the right of area 2 with a single manoeuvre. Given that a trajectory from the first starting point must pass to the right of area 2, area 1 does not represent a real constraint. The left-most trajectory is not of interest: forbidden areas are only of interest when they get in the way - we are not interested in visiting them for their own sake.

The principle of only considering trajectories which are constrained by forbidden areas is termed 'rubber-banding', by analogy with a rubber-band which, when threaded between obstacles, only touches constraining obstacles once stretched. This principle was used in the ARC project[2].

Within the GEARS algorithm the rubber-banding principle can be applied very simply by limiting the range of turn angles within which the search of avoiding manoeuvres is performed, from a given starting point.

Considering solution trajectories consisting of manoeuvres of a single type, the application of rubber-banding allows a reduction on average by half in the number of avoiding manoeuvres found from a starting point. For a tree of depth n, this results in a reduction in the number of nodes in the tree by a factor of about 2^n. For long look-ahead times, which would give rise to deep trees, application of this principle allows an enormous reduction in the size of the tree to be constructed. A longer discussion of rubber-banding is given elsewhere.[8]

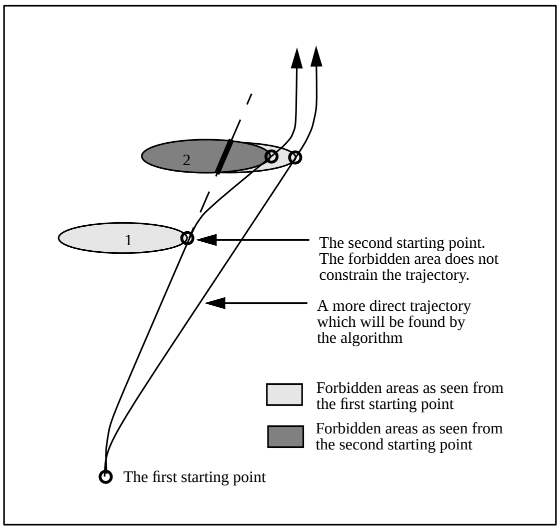

Fig. 10: Rubber-banding

2.10 Vertical and Speed Change Avoiding Manoeuvres

The first step in the resolution algorithm is the generation of a preferred manoeuvre. In non-trivial cases the preferred manoeuvre will bring the manoeuvred aircraft into conflict with one or more obstacle aircraft. A set of possible modifications to the preferred manoeuvre is then considered. In the case of lateral resolution, the manoeuvred aircraft turns from its current heading to a new heading. However, other sets of possible modifications, each indexed by some parameter, can be considered. For example, the manoeuvred aircraft might climb or descend from its current flight level to reach a new flight level or accelerate or decelerate from its current speed to reach a new speed. In all cases a transition is made from an initial value of the indexing parameter to a final value and the final value is then maintained.

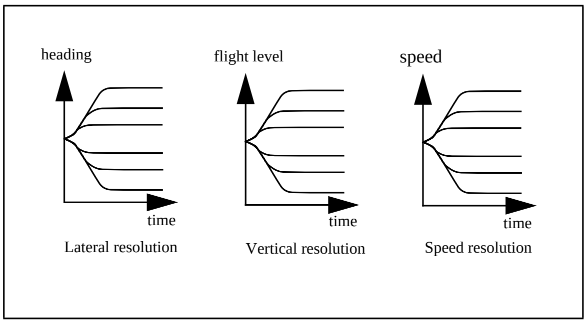

Fig. 11: Sets of Possible Modifications

A candidate manoeuvre is found by increasing or decreasing the value of the indexing parameter until a manoeuvre is found which allows the manoeuvred aircraft to avoid the given obstacle aircraft. The trajectory change point at which the manoeuvred aircraft may attempt to follow its preferred manoeuvre is then determined.

The algorithm for finding avoiding manoeuvres effects a search in a space defined by an indexing parameter and time. It is applicable regardless of whether the indexing parameter is heading, flight level or speed.

2.11 Single Manoeuvre Type and Mixed Manoeuvre Trajectories

At successive trajectory change points avoiding manoeuvres may be found by always considering only a given type of modification, for example, only lateral modifications or only vertical modifications.

Alternatively, avoiding manoeuvres can be found by considering more than one type of modification at each trajectory change point, for example, lateral and vertical modifications, resulting in solution trajectories in which successive manoeuvres may have been obtained by different types of modification. However, the rubber-banding principle cannot be applied in this case.

More than one set of possible modifications could be defined for a given modification type. For example, different sets of modifications might correspond to different rates of transition from the initial value to the final value, *e.g.* different rates of climb. If no acceptable solution trajectories were found using one set of possible manoeuvres the search could be repeated using a different set.

3. Illustrations from Prototype

The following illustrations were created using a prototype which performed lateral resolution. The manoeuvre generator assumes that aircraft perform standard turns *i.e.* they have a turn radius of 22.5 nautical miles. The conflict detector implements a simple, time-dependent separation standard. Initially a separation of 5 nautical miles is required. This value grows at 30 knots to allow for uncertainty in the predicted position of each aircraft growing at 15 knots in all directions, or 0.25 nautical miles per minute*. To simplify interpretation of the illustrations all aircraft have the same speed (360 knots) and they all have the same vertical profile (height as a function of time).

3.1 Individual Motion Planning

The first illustration (Fig. 12) shows the current positions and predicted tracks of a number of aircraft. The aircraft to be manoeuvred is the leftmost one (labelled ‘a’) and its destination is to the right. The black separation disc around each aircraft has a diameter of 5 nautical miles so that if two discs were to touch the aircraft would be just separated, at the start of the resolution period. The forbidden areas due to the obstacle aircraft at this time are shown solely in order to better visualise the underlying problem. The calculation of these areas does not constitute a step in the algorithm.

The second illustration (Fig. 13) shows the avoiding manoeuvres found from the starting point. The search for avoiding manoeuvres uncovers only those forbidden areas and parts of forbidden areas which are visible from the starting point and which lie across candidate manoeuvres considered earlier in the search.

The third illustration in this sequence (Fig. 14) shows the complete tree of avoiding manoeuvres consisting of five (first-class), conflict-free trajectories. (Forbidden areas are not shown here as their exact positions and shapes depend upon the turning points from which they are viewed). Individual planning corresponds well to free-flight and free-route scenarios in

* An along-track error growth rate of 0.25 nautical miles per minute is reported by Paeilli and Erzberger[7]. The across-track error is approximately constant with a magnitude of less than 0.5 nautical miles. The separation standard used here is for illustrative purposes only and was chosen for ease of implementation. Use of the same rate of growth of error across-track as along-track is unduly pessimistic.

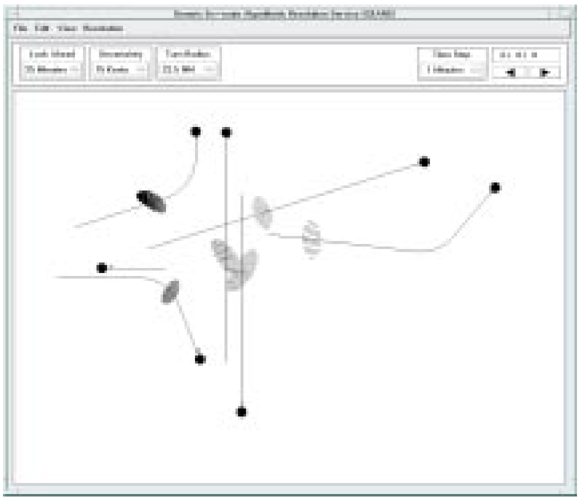

Fig. 12: Forbidden Areas at Start of Resolution

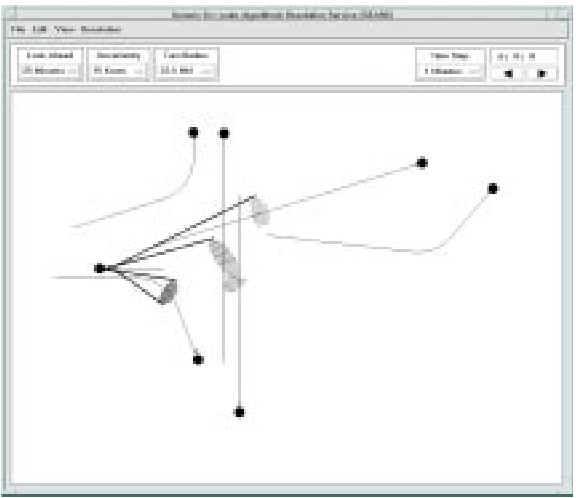

Fig. 13: The Avoiding Manoeuvres from the Starting Point

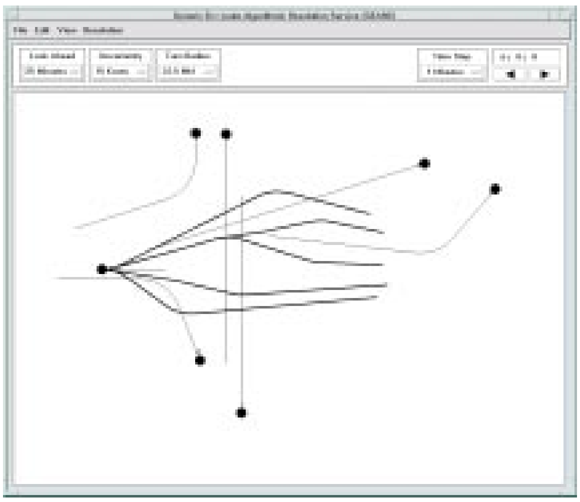

Fig. 14: The Set of Flyable, Conflict-Free Trajectories

which in a given situation it is the responsibility of one aircraft to manoeuvre in order to avoid other aircraft.

3.2 Prioritized Planning

The next illustration (Fig. 15) shows the result of applying the resolution algorithm to each of ten aircraft arranged in a circle, which would otherwise enter into a simultaneous conflict in the centre of the circle. The circle has a diameter of 150 nautical miles (which corresponds to 25 minutes flying time at 360 knots). For each aircraft the solution selected from the set is the one which takes the aircraft nearest to its destination at the end of the resolution time-interval. An arbitrary order of priority is used: solutions are found in sequence going clockwise around the circle, beginning with aircraft number 1 and ending with aircraft number 10. (This situation is a good test that the algorithm copes with overlapping forbidden areas).

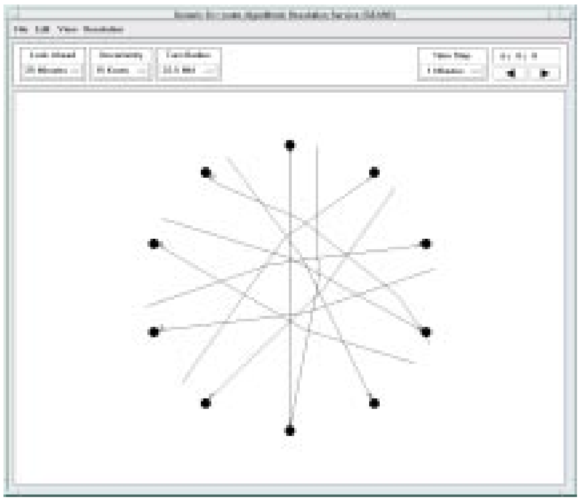

Fig. 15: Conflict-Free Trajectories for Ten Aircraft

The final illustration in this section (Fig. 16) shows the predicted positions of the aircraft when they will be most tightly packed, 12.5 minutes after the resolution. The required separation at this time is 11.25 miles. It can be seen that the separation discs around each aircraft have grown, but that none of the discs overlap.

Prioritized planning corresponds well to free-flight and free-routing scenarios in which, in a given situation, it is the responsibility of a controller to replan the trajectories of more aircraft than one aircraft. If the algorithm were to be used to provide conflict resolution assistance to a controller, one can imagine that he or she might select from a traffic situation those aircraft whose trajectories are to be replanned and then replan them one at a time in an order of his or her choosing.

4. Computational Complexity

In discussions of computational complexity it is customary to begin by considering worst cases. Considering only a single type of modification of the preferred manoeuvre, e.g. lateral-only or vertical-only resolution, for N forbidden areas the worst case (when planning trajectories for a single aircraft) arises when

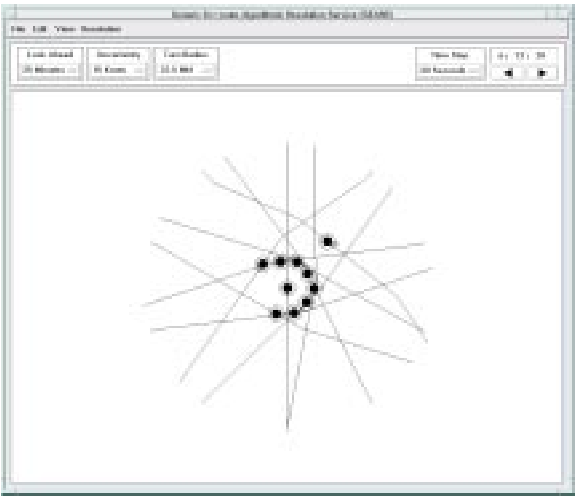

Fig. 16: The Situation when the Aircraft are Most Tightly Packed

each forbidden area doubles the total number of solutions so that the total computation required grows as 2^N. In mixed-manoeuvre resolution with m manoeuvres avoiding each area, the total computation required grows as m^N, *i.e.* exponentially. Of course the worst case would require careful alignment and spacing of obstacle aircraft.

A number of points should be borne in mind when considering the efficiency of such an algorithm either in air traffic control simulations or eventually operationally. Firstly, the rate of growth of uncertainty in the predicted positions of aircraft is, in practice, sufficiently high that there may be little value in trying to solve a potential conflict a long time in advance: the conflict may well not occur. Secondly, foreseeable traffic densities (*e.g.* a doubling of traffic by 2015) are such that the number of forbidden areas seen by the algorithm during a realistic look-ahead period is typically quite small. In dense situations forbidden areas tend to be hidden partially or completely by other areas and may well overlap, the effect being to reduce the total number of trajectories found by the algorithm.

In the case of 10 aircraft converging on a point shown earlier the time taken per resolution for aircraft numbers 7 - 10 (potentially conflicting with 6 - 9 other aircraft) is about 6 seconds on a Unix work-station (HP 9000 model 755). In the implementation of the prototype little effort was made to minimise execution time. The performance of the resolution algorithm in terms of time taken is highly dependent on that of the conflict detector and manoeuvre generator.

5. Use of the Algorithm in MAICA Simulations

The European Commission funded a project concerned with the Modelling and Analysis of the Impact of Changes in Air Traffic Management (MAICA). As part of this project Alcatel ISR developed the MARS simulator (Multiple Autonomous aiRcraft Simulator) which was then used to perform simulations, with the purpose of investigating the feasibility of autonomous aircraft operations. The results of these simulations are presented in full in the MAICA Final Report[9]. The GEARS algorithm was used in these simulations, and a brief overview of the performance of the resolution strategy is presented here.

The main underlying assumptions were that all aircraft involved would be equipped with ADS-B (Automatic Dependent Surveillance - Broadcast) with a range of 150 nautical miles, capable of broadcasting future trajectories with a look-ahead time of 10 minutes, together with a flight management system (FMS) capable of maintaining the aircraft on a given 4-dimensional trajectory. A potential conflict was detected whenever the trajectories of two aircraft (in ADS-B range) would lead to them being separated by less than 5 nautical miles and 1000 feet. Traffic samples were derived from current traffic with traffic levels up to three times current peak levels.

Following detection of a conflict the resolution strategy used in these simulations was to apply a proposed set of Extended Flight Rules[10] to designate one of the two aircraft involved in the conflict to be responsible for its resolution and then to use the GEARS algorithm to generate a conflict-free trajectory for that aircraft. The key constraint of this strategy is that only one trajectory is modified in order to re-establish a conflict-free situation. Resolutions were performed instantaneously (in simulated time) within a range of times following conflict-detection. Solution trajectories became known immediately to all aircraft in ADS-B range. A realistic aircraft model was used for manoeuvre generation and the rubber-banding heuristic was applied throughout.

Seventy-six simulations were performed varying geographical region and traffic level. A look-ahead time of ten minutes was used and the algorithm was used to generate mixed-manoeuvre trajectories. In seventy-two of these simulations solution trajectories were always found for the designated aircraft on the first call to the resolution algorithm (over 17,000 resolutions). In the four remaining simulations ten cases of missed resolutions were observed, that is, conflicts which were not solved on the first call to the algorithm but which were solved on a subsequent simulation step. (Presumably other changes in the traffic situation made resolution possible at a later time). One conflict was not solved either on the first call nor on subsequent calls. The four simulations in which anomalous cases occurred included the core area of Europe. Ten of the eleven anomalous cases occurred in three simulations in which the traffic level corresponded to about three times current peak traffic levels. During these three simulations alone over 17,000 resolutions were performed, and the probability of successful resolution on the first call exceeded 99.9% in each simulation. Nonetheless, the existence of anomalous cases indicates a limitation of the resolution strategy. Evaluation of the resolution strategy was not the primary goal of the MAICA project and it was not possible within the timeframe of the project to further investigate these cases.

The average number of obstacle aircraft seen by the algorithm during these simulations was typically quite low (between one and three). The maximum number of obstacle aircraft taken into account in a resolution was nineteen.

A further five simulations investigated the influence of a decreased look-ahead time on the number of conflicts detected. Three more simulations investigated the consequences of differently weighted preferences for lateral, vertical and mixed manoeuvre resolutions.[9]

6. Conclusions

A conflict resolution algorithm has been described

- which generates a set of solution trajectories which are necessarily conflict-free
- which is applicable with a wide range of aircraft behaviour models (if the aircraft behaviour model is sufficiently realistic the solution trajectories will be 'flyable').
- which is applicable with a wide range of definitions of 'conflict' including time- and geometry-dependent definitions, allowing uncertainty to be modelled
- which can perform several types of resolution e.g. lateral-only, vertical-only, speed-only as well as mixed-manoeuvre resolution
- which can be used for individual planning of the trajectory of one aircraft
- which can be used for prioritized planning of the trajectories of many aircraft
- and whose operation is straightforward to understand and implement

This approach to resolution allows a decomposition into independent and well-known sub-problems, namely manoeuvre prediction (a very restricted form of trajectory prediction) and conflict detection. If this approach is adopted the focus of development shifts to the development of appropriate conflict detectors and manoeuvre generators.

The algorithm has been used in simulations investigating the feasibility of autonomous aircraft operations and might be used in operational applications (ground or air-based) if, due to time or workload pressures, conflict-resolution assistance is required. It may also have applications outside of air traffic control.

Acknowledgements

I would like to thank Nicolas Gautier for innumerable discussions which greatly contributed to the development of the algorithm and who worked with me on the implementation of the prototype. Bernard Maudry, who introduced the concept of forbidden area and who developed the lateral resolution algorithm used in the ARC project, explained the difficulties which were encountered and pointed out the applicability of the rubber-banding principle. Pierre Faure of Alcatel ISR developed the MARS simulator used in the MAICA simulations and implemented a version of the GEARS algorithm allowing lateral, vertical and mixed-manoeuvre resolution. Karim Zeghal suggested a number of improvements to the text.

References

[1] Erdmann M. and Lozano-Perez T. "On Multiple Moving Objects", IEEE International Conference on Robotics and Automation, pp. 1419 - 1424, 1986.

[2] ARC 2000 Team, "ARC 2000 Technical Report", EEC Report 274, Eurocontrol Experimental Centre, December 1994.

[3] Whysall P. and Cordingley S., "Specification of the ARC2000 Conflict Resolution Algorithm", Roke Manor Research Ltd., Report No. 72/95/R/113/U, August 1995.

[4] Silbert M., "An Efficient Algorithm for Path Planning to Avoid Dynamic 3D Obstacles", Proceedings of Associate Technology Conference, 1991, pp. 193-204.

[5] Silbert M., "Comparison of the Event-Step Algorithm to Other Path Planning Methods to Avoid Dynamic 3D Obstacles", AGARD Conference Proceedings 504, October 1991, pp 14-1 - 14-9.

[6] Kuchar, J.K. and Yang L.C., "Survey of Conflict Detection and Resolution Modelling Methods", AIAA-97-3732, AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, August 11-13, 1997.

[7] Paielli R.A. and Erzberger H., "Conflict Probability Estimation for Free Flight", AIAA Journal of Guidance, Control and Dynamics, Vol.20, Number 3, May-June 1997, pp 588 - 596.

[8] Irvine R., "The GEARS Conflict Resolution Algorithm", EEC Report No. 321, Eurocontrol Experimental Centre, November 1997.

[9] Faure P. et al., MAICA Final Report, MAICA:AED-11-DOC-WP4-R2.0, 2nd edition, Aerospatiale, 15th May 1998

[10] Duong N., Hoffman E., Nicoleon J-P., "Initial Results of Investigation into Autonomous Aircraft Concept", Proceedings of 1st FAA/Eurocontrol ATM R&D Seminar, Saclay, France, June 1997.