

Technical University of Crete

**Conflict Avoidance under Free Flight regime
using NonLinear Mixed-Integer Programming**

By

Nikos Papathomopoulos

A Diploma thesis submitted in partial fulfillment of the requirements
of the degree in Electronic and Computer Engineering

Approved by _____

Chairperson of the supervisory committee

Manolis Christodoulou _____

Michalis Zervakis _____

Emmanouil Varvarigos _____

Chania 2003

Contents

1	Introduction to Air Traffic Management.....	7
1.1	A brief reference to ATM through the years.....	7
1.2	System Overview.....	8
2	The concept of Free Flight.....	12
2.1	What is Free Flight?.....	12
2.1.1	A first definition.....	12
2.1.2	System Elements and Functions.....	14
2.2	System Level Issues.....	16
2.3	Human Factor Issues.....	18
3	Proposed Approach.....	20
3.1	Framework.....	20
3.2	Problem Statement.....	21
3.3	Problem Formulation.....	23
3.3.1	The VC problem.....	23
3.3.2	The HAC problem.....	28
3.4	The Combined Approach.....	31
3.4.1	The Final Set of Conflict Constrains for Combined Approach.....	31
3.4.2	Formulation of the Final Conflict Constrains into MIP Constrains.....	34
3.5	Simulation and Case Scenarios.....	37
3.5.1	Case Scenario 1.1($z=z_1$).....	41
3.5.2	Case Scenario 1.2($z=z_1$).....	43
3.5.3	Case Scenario 1.3 (head to head)($z=z_1$).....	45
3.5.4	Case Scenario 1.4($z=z_1$).....	47
3.5.5	Case Scenario 1.5($z=z_1$).....	49
3.5.6	Case Scenario 1.6($z=z_1$).....	51
3.5.7	Case Scenario 1.7($z=z_1$).....	53
3.5.8	Case Scenario 1.8($z=z_1$).....	55
3.5.9	Case Scenario 2.1($z=z_2$).....	57
3.5.10	Case Scenario 2.2($z=z_2$).....	59
3.5.11	Case Scenario 2.3($z=z_2$).....	61
3.5.12	Case Scenario 2.4($z=z_2$).....	63
3.5.13	Case Scenario 2.5($z=z_2$).....	65
3.5.14	Case Scenario 2.6($z=z_2$).....	67
3.5.15	Case Scenario 2.7($z=z_2$).....	69
3.5.16	Case Scenario 2.8($z=z_2$).....	71
3.6	Conclusive Results.....	73

Appendices.....	75
A Basic Features of GAMS.....	75
A.1 General Principles.....	75
A.2 Documentation.....	76
A.3 Portability.....	76
A.4 User Interface.....	77
A.5 Model Library.....	77
B Mixed Integer Non-Linear Programming.....	78
B.1 Introduction.....	78
B.2 Algorithms.....	79
B.2.1 Solution Approaches.....	80
B.3 Software.....	80
B.3.1 Commercial MINLP Software Packages.....	81
B.3.2 Modelling Languages.....	81
B.4 Recent Developments.....	82
B.5 Future Directions.....	82
C Software Codes and Relative Documentation.....	84
References.....	93

List of Figures

- 1.1 The Hierarchical Nature of Air Traffic Management
- 1.2 Representation of Air Traffic Management Organizational Structure
- 3.1 Geometric construction for conflict avoidance constrains in the case of intersecting trajectories of the VC problem. In this case Aircraft 1 do not intersect the shadow generated by Aircraft 2 then no conflict will occur between the two aircraft.
- 3.2 The two non parallel straight lines tangent to the safety disks of radius $d/2$ for two aircraft at distance $A_{12} / 2$.
- 3.3 Geometric construction for conflict avoidance constrains in the case of intersecting trajectories for the HAC problem. In this case Aircraft 1 intersect the shadow of Aircraft 2, then a future conflict has been detected.
- 3.4 Example of forbidden sectors in the Los Angeles control sector. For the aircraft A we have to introduce more constrains on the direction of flight due to forbidden zones of airspace.
- 3.5 Initial and final configuration points of two aircrafts traversing the control circle.
- 3.6 A comprehensive example figure about the aircrafts' behaviour before and after conflict resolution.
- 3.7 Scenario 1.1's distributed aircrafts and their trajectories before conflict resolution($z=z_1$).
- 3.8 Scenario 1.1's distributed aircrafts and their trajectories after conflict resolution($z=z_1$).
- 3.9 Table showing the velocities, heading angles and overall deviations for scenario 1.1's distributed aircrafts. CR stands for Conflict Resolution.
- 3.10 Scenario 1.2's distributed aircrafts and their trajectories before conflict resolution($z=z_1$).
- 3.11 Scenario 1.2's distributed aircrafts and their trajectories after conflict resolution($z=z_1$).
- 3.12 Table showing the velocities, heading angles and overall deviations for scenario 1.2's distributed aircrafts. CR stands for Conflict Resolution.
- 3.13 Scenario 1.3's distributed aircrafts and their trajectories before conflict resolution($z=z_1$).
- 3.14 Scenario 1.3's distributed aircrafts and their trajectories after conflict resolution($z=z_1$).
- 3.15 Table showing the velocities, heading angles and overall deviations for scenario 1.3's distributed aircrafts. CR stands for Conflict Resolution.

- 3.16 Scenario 1.4's distributed aircrafts and their trajectories before conflict resolution($z=z1$).
- 3.17 Scenario 1.4's distributed aircrafts and their trajectories after conflict resolution($z=z1$).
- 3.18 Table showing the velocities,heading angles and overall deviations for scenario 1.4's distributed aircrafts.CR stands for Conflict Resolution.
- 3.19 Scenario 1.5's distributed aircrafts and their trajectories before conflict resolution($z=z1$).
- 3.20 Scenario 1.5's distributed aircrafts and their trajectories after conflict resolution($z=z1$).
- 3.21 Table showing the velocities,heading angles and overall deviations for scenario 1.5's distributed aircrafts.CR stands for Conflict Resolution.
- 3.22 Scenario 1.6's distributed aircrafts and their trajectories before conflict resolution($z=z1$).
- 3.23 Scenario 1.6's distributed aircrafts and their trajectories after conflict resolution($z=z1$).
- 3.24 Table showing the velocities,heading angles and overall deviations for scenario 1.6's distributed aircrafts.CR stands for Conflict Resolution.
- 3.25 Scenario 1.7's distributed aircrafts and their trajectories before conflict resolution($z=z1$).
- 3.26 Scenario 1.7's distributed aircrafts and their trajectories after conflict resolution($z=z1$).
- 3.27 Table showing the velocities,heading angles and overall deviations for scenario 1.7's distributed aircrafts.CR stands for Conflict Resolution.
- 3.28 Scenario 1.8's distributed aircrafts and their trajectories before conflict resolution($z=z1$).
- 3.29 Scenario 1.8's distributed aircrafts and their trajectories after conflict resolution($z=z1$).
- 3.30 Table showing the velocities,heading angles and overall deviations for scenario 1.8's distributed aircrafts.CR stands for Conflict Resolution.
- 3.31 Scenario 2.1's distributed aircrafts and their trajectories before conflict resolution($z=z2$).
- 3.32 Scenario 2.1's distributed aircrafts and their trajectories after conflict resolution($z=z2$).
- 3.33 Table showing the velocities,heading angles and overall deviations for scenario 2.1's distributed aircrafts.CR stands for Conflict Resolution.
- 3.34 Scenario 2.2's distributed aircrafts and their trajectories before conflict resolution($z=z2$).
- 3.35 Scenario 2.2's distributed aircrafts and their trajectories after conflict resolution($z=z2$).
- 3.36 Table showing the velocities,heading angles and overall deviations for scenario 2.2's distributed aircrafts.CR stands for Conflict Resolution.

- 3.37 Scenario 2.3's distributed aircrafts and their trajectories before conflict resolution($z=z_2$).
- 3.38 Scenario 2.3's distributed aircrafts and their trajectories after conflict resolution($z=z_2$).
- 3.39 Table showing the velocities,heading angles and overall deviations for scenario 2.3's distributed aircrafts.CR stands for Conflict Resolution.
- 3.40 Scenario 2.4's distributed aircrafts and their trajectories before conflict resolution($z=z_2$).
- 3.41 Scenario 2.4's distributed aircrafts and their trajectories after conflict resolution($z=z_2$).
- 3.42 Table showing the velocities,heading angles and overall deviations for scenario 2.4's distributed aircrafts.CR stands for Conflict Resolution.
- 3.43 Scenario 2.5's distributed aircrafts and their trajectories before conflict resolution($z=z_2$).
- 3.44 Scenario 2.5's distributed aircrafts and their trajectories after conflict resolution($z=z_2$).
- 3.45 Table showing the velocities,heading angles and overall deviations for scenario 2.5's distributed aircrafts.CR stands for Conflict Resolution.
- 3.46 Scenario 2.6's distributed aircrafts and their trajectories before conflict resolution($z=z_2$).
- 3.47 Scenario 2.6's distributed aircrafts and their trajectories after conflict resolution($z=z_2$).
- 3.48 Table showing the velocities,heading angles and overall deviations for scenario 2.6's distributed aircrafts.CR stands for Conflict Resolution.
- 3.49 Scenario 2.7's distributed aircrafts and their trajectories before conflict resolution($z=z_2$).
- 3.50 Scenario 2.7's distributed aircrafts and their trajectories after conflict resolution($z=z_2$).
- 3.51 Table showing the velocities,heading angles and overall deviations for scenario 2.7's distributed aircrafts.CR stands for Conflict Resolution.
- 3.52 Scenario 2.8's distributed aircrafts and their trajectories before conflict resolution($z=z_2$).
- 3.53 Scenario 2.8's distributed aircrafts and their trajectories after conflict resolution($z=z_2$).
- 3.54 Table showing the velocities,heading angles and overall deviations for scenario 2.8's distributed aircrafts.CR stands for Conflict Resolution.
- 3.55 Table showing the absolut sum of the velocity and heading angle deviations for z_1 and z_2 simulations.

Abstract

In the early days of aviation there was little need for a ground based air traffic control, because of the small amount of aircraft that shared the aerospace. But as the years went by, it became apparent that some rules have to be imposed, a ground based control system was necessary in order to have safe and efficient flights. Nevertheless, in the present time, the traffic confusion in the skies became intolerable and a need emerged, towards the idea of **free flight**, which in a few words allows real time changes by the operators in the aircrafts' nominal paths and speed, to suit potential crisis situations in the air (such as possible conflict, bad weather, etc.).

In this essay, we consider the problem of solving conflicts arising among two aircraft that are assumed to move in a shared airspace, under free flight regime. Aircraft can not get closer to each other than a given distance in order to avoid possible conflicts between them. We are particullary interested in finding optimal non-conflict paths for both aircraft (if a conflict has been detected), in a sence of minimizing their deviation from their originally scheduled path. In Air Traffic Control literature, two different methods of the conflict resolution problem have been considered, which can be formulated as mixed-integer linear problems. In the first only velocity changes are admissible maneuvers and in the second one only heading angle changes are allowed. We will propose a combined approach of the above mentioned methods which will allow both velocity and heading angle deviations. This proposed approach will be formulated as a mixed-integer nonlinear problem, and solutions will be obtained in a few seconds with the help of a standard optimization tool.

In conclusion, we can say that the proposed approach is much more efficient in real life situations than the methods introduced in ATC literature, just because it allows two kind of maneuver. Although we are certain that in this case computational time is greater, due to the nonlinearity, the chosen optimization tool produces the results in a matter of a few seconds, thus we are optimistic that this approach may be used as part of a real or fast-time simulation.

Chapter 1

Introduction to Air Traffic Management

1.1 A brief reference to ATM through the years

Prior to the early 1930's there was little need for an organised system of air traffic management(ATM).The original methods of air traffic controlling was for someone to stand in a prominent position on the airfield and use coloured flags. If the controller waved a green flag, it meant that the pilots could proceed with their planned takeoff or landing.But if a red flag was waved then the pilots would have to hold their position until the controller determined that it was safe to continue. In the late 1930's a new method had been developed whereby the controller would use a light gun to direct coloured beam's of light directly at the intended aircraft. The colours used resembled the fag system, but worked a lot better as is could be used at night and signals could be directed at specific aircraft without any chance of confusion between multiple aircraft.

Development of the ATM system after World War II was rapid. It incorporated new technologies in radar, communications, aircraft, and engines and capitalized on an expanding economy and a more mobile population. However, the system was constrained by the radar, communications, and navigation technologies. It was a network of “highways in the sky” in which aircraft followed prescribed routes with monitored vertical and horizontal tolerances.

Differences in aircraft performance were accommodated in this linear system by aircrew negotiation with air traffic controllers or by the air traffic controllers themselves. The diversity of participating parties has greatly complicated the efficiency and congestion of the system. In the 1960s, development of inertial navigation systems provided a new capability and resulted in a new area navigation category.Equipped aircraft and trained crews

could make their own routes, but they still had to follow standard departure and arrival routes and procedures. This version of the ATM system, with minor modifications and decades-old computer technology, has evolved into the current system that is a centralized, ground-based positive ATM system.

1.2 System overview

The design of the ATM system reflects important concepts about information management and social organization. Figure 1.1 illustrates the basic hierarchical nature of this design and of the management domains that make up the air traffic environment. Information regarding air traffic can be partitioned into a hierarchy of domains based on its quality and granularity. As is shown in the figure, long-run strategic planning based on aggregate traffic demand data must be done to make decisions about resource allocations and about rules and procedures, along with daily strategic traffic flow management decisions, while tactical activities, based on flight-specific data, must be done to assure separation of individual aircraft. The goal of these strategic planning activities is to ensure that controllers, pilots, and dispatchers can safely coordinate the activities of specific aircraft in order to assure safe, effective utilization of the airspace and airport facilities.

The ATM system is also organized hierarchically to take advantage of the informational structure (see Figure 1.2). At an organizational level, the ATM system has a hierarchical structure in which the tactical and strategic air traffic management functions are allocated among various ATC and TFM(Traffic Flow Management) organizational units. Although it is useful to discuss these functions as if they were associated with discrete organizational units, in practice the elements of the functions are intertwined with overlapping and redundant responsibilities to ensure system reliability.

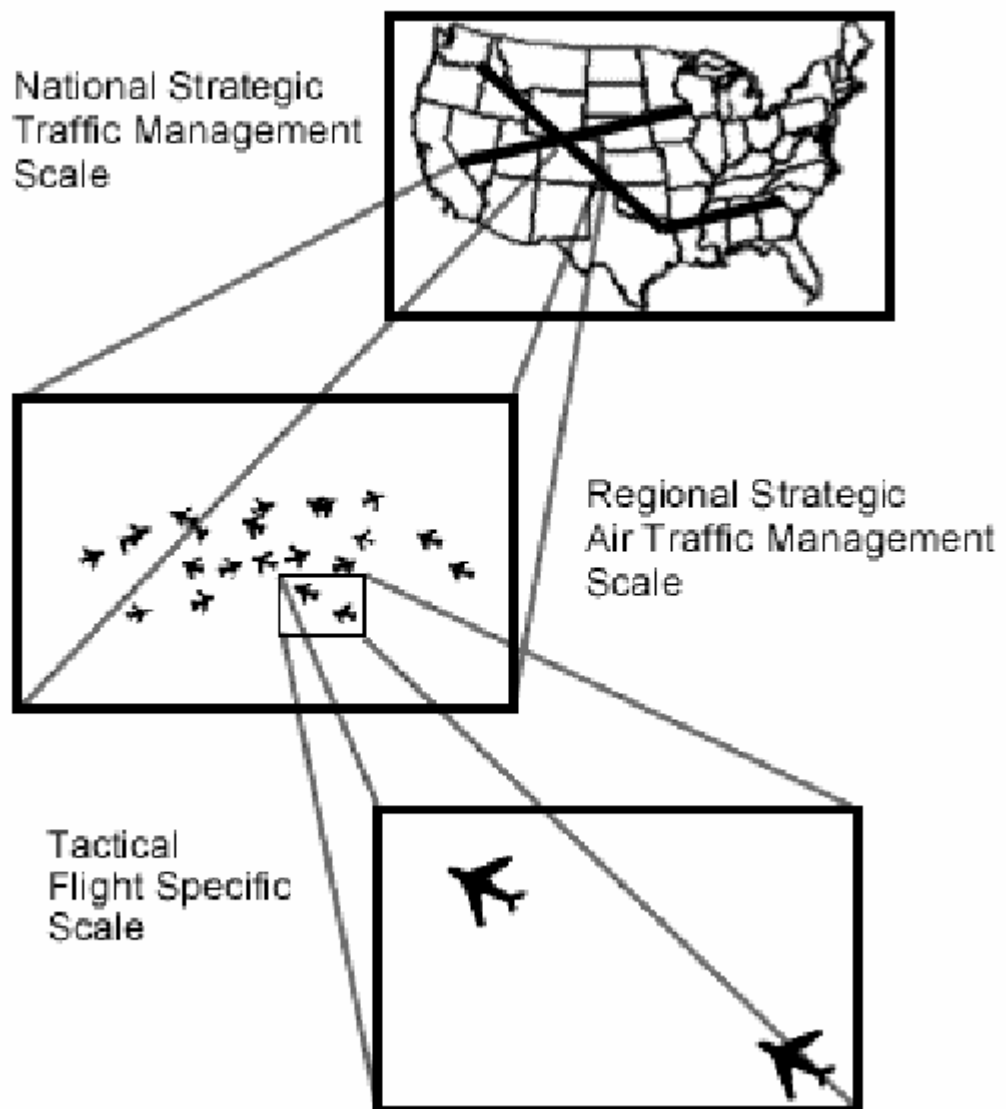


Figure 1.1. The hierarchical nature of Air Traffic Management

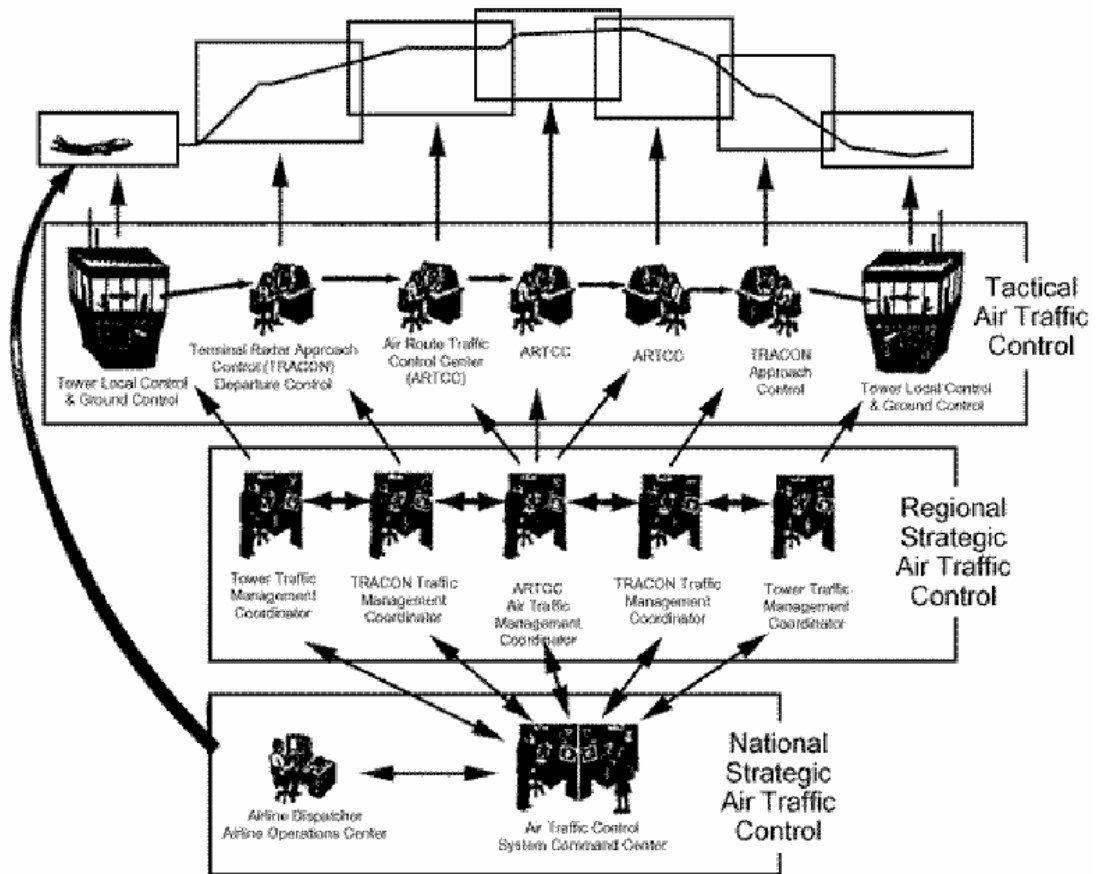


Figure 1.2. Representation of Air Traffic Management Organizational Structure

At the lowest level of organization, the ATM system is organized to assure aircraft separation throughout all phases of flight from takeoff to landing. To accomplish this, the ATC component of the system is composed of controllers and specialists working in several different types of facilities. These include:

- a.** Towers at airports, with air traffic controllers responsible for directing arrivals and departures.
- b.** Terminal radar-approach control facilities (TRACONs), with controllers guiding flights for roughly the first and last 40 miles of flight.
- c.** Air route traffic control centers (ARTCCs) with controllers in charge while en route.
- d.** Flight service stations (FSSs) with flight service specialists providing services such as flight plan filing, preflight and inflight weather briefings to general aviation pilots.

Pilots file a flight plan and obtain a clearance in order to fly through airspace controlled by the system. Once the initial clearance has been obtained, the pilot maintains radio communication with the controllers at these facilities to receive ATC services. The controllers monitor specific flights within the airspace (sectors) that they are in charge of and issue instructions to pilots in order to ensure safe separation and efficient use of the airspace.

At an intermediate level of organization, the ATM system includes local TFM units. Each ARTCC and TRACON also has a traffic management unit and there are a few towers where traffic management coordinator positions have been established. These organizations are responsible for helping to plan and adjust the flow of traffic within their airspace. At the national level, a centralized facility, the Air Traffic Control Systems Command Center (ATCSCC) coordinates the activities of the local units. Finally, the dispatchers within airline operations centers (AOCs) have an increasingly significant impact on traffic flows, and thus must be considered in any discussion of the system.

Chapter 2

The concept of free flight

2.1 What is free flight?

2.1.1 A first definition

”A safe and efficient flight operating capability under instrument flight rules (IFR) in which the operators have the freedom to select their path and speed in real time. Air traffic restrictions are only imposed to ensure separation, to preclude exceeding airport capacity, to prevent unauthorized flight through special use airspace, and to ensure safety of flight. Restrictions are limited in extent and duration to correct the identified problem. Any activity which removes restrictions represents a move toward free flight.”

The above statement has been developed by a working committee on free flight sponsored by the RTCA[1] . In a few words we can say that free flight is a new way of managing air traffic that was originally designed to enhance safety, capacity and efficiency of the U.S. NAS(National Aerospace System). Under this new management system air traffic control is expected to move gradually from a highly structured system based on elaborate moves and procedures to a more flexible system that allows pilots, within limits, to change their route speed and altitude, notifying the air traffic controller of the new route. But how does this situation affect all the separate parts of air traffic management?

The national airspace involves four key components: (1) air traffic control personnel, (2) dispatchers and management of airline industry, (3) pilots and their aircraft systems, (4) ground-based automation. Here we take a systems perspective, in considering the roles of these four components in the concept of free flight. We begin by characterizing differences among the key components in three critical variables:

1. GOALS may differ, in terms of the relative emphasis on safety versus efficiency (and productivity) and in terms of local optimization versus global optimization.
2. INFORMATION may reside differently within different components and such information may or may not be shared between them.
3. AUTHORITY for different aspects of air traffic management exists in certain places. Furthermore authority may “flow” along certain paths and these paths may change with future changes in air traffic management procedure.

First, air traffic controllers maintain primary responsibility for the goal of overall safety of all aircraft in the system, and their concerns about efficiency are distributed across all occupants of the airspace. Airline management, as reflected by the influence of the airline operation center, although concerned with safety, has relatively greater interest in expediency and efficiency, as well as a more local interest in the efficiency of its own fleet of aircraft. Profit is a heavy driver of the expediency goal, given the low profit margin of most airlines and the high cost of delay to company profit. The pilot's interests are still more local, concerned primarily with safety and expediency of a single aircraft. Automation may be conceived to be relatively goal-neutral with regard to safety and efficiency, in that these goals are defined by the designers of the system. However, many aspects of automation proposed for the national airspace system are specifically intended to increase efficiency, with the explicit requirement that they be safety-neutral. The fact remains that automation may sometimes be safety-compromising if it is not carefully implemented.

Second, each component retains some what different information about the airspace. That information can vary in its geographical scope (global to local), its level of detail, and its accuracy. For example, airline dispatchers and management at the airline operations department may currently possess the best information about global weather patterns of regions containing its flying fleet. Relatively high levels of automation can provide them with accurate projections of ideal flight routes. Controllers and towers have more restricted but detailed information and pilots generally have the most restricted but detailed information regarding their capabilities and intent of their own aircraft. Thus, across these components, there tends to be a trade-off between information scope and detail. An advantage of automation is that it has the ability to retain, digest and share information that is at once global and detailed and thus to contribute in a beneficial way to information sharing.

Third, the Federal Aviation Administration (FAA) has set up clearly defined lines of actual authority (responsibility) for different aspects of flight path management. For example, controllers have authority to issue clearances and instructions to aircraft only within their sector. Controllers, not pilots, have authority to direct instrument flight rules aircraft to different flight levels and headings. However, it is not clear that perceived authority necessarily follows procedurally defined authority lines. For example, the possible loss of situation awareness and skill induced by high levels of flight deck automation can create a potential shift in perceived authority for trajectory management away from the pilot. If the automation is trusted, reliable, and introduced carefully into the workplace, this shift can be voluntary (i.e. the human can willingly give up some aspects of control to automation). However, if the automation is mistrusted, clumsy, and introduced without consideration of user inputs, the shift may be involuntary, with the user feeling that authority has been taken away. In either case, there are possible concerns: complementarity in the former case, loss of job satisfaction or even possible abuse of the automation in the latter.

From a controller's perspective, the loss of authority and information may have similar implications, no matter which component in the airspace (pilot or automation) is perceived to preempt that authority. In general, a scenario is addressed in which authority potentially flows to automation. However, the concept of free flight in which pilots, airline dispatchers and managers rather than automation may be assuming more authority, has many implications for the controller.

2.1.2 System Elements and Functions

The numerous versions of proposed free flight architectures have in common a set of key elements.

Global Positioning System and Position Broadcasting

Any aircraft must have a very accurate estimate of own position and that of its nearest neighbours. The global positioning system appears to provide this facility and, when coupled with automatic dependent surveillance, will enable rapid communications of accurate navigation information between aircraft in close proximity. Such information can also be broadcast to air traffic and airline operation centers.

Traffic Display

In order to plan conflict-free trajectories and to maneuver around possible conflicts in the absence of air traffic control advisories, pilots will need an accurate cockpit display of traffic information. Such displays include an important distinction between conflict or protected zone, the region of space that would formally define a loss of separation or operational error, and an alert zone. The latter would be the level of separation at which an advisory to maneuver would be offered to one or both aircraft. It may also define a time at which air traffic control might be alerted to the possibility that active control from the ground might be required. Since the parameter dictating the degree of urgency to maneuver is the predicted time to contact (rather than spatial separation) many have considered the alert zone to be time based rather than space based and hence not simply represented in its geometrical form.

Intent Inferencing

Any traffic display designed to alert the pilot to potential conflicts will be beneficial to the extent that it can account for reliable predictive information regarding the trajectory of both aircraft involved. Accounting for the current velocity and acceleration vector provides a good deal of accuracy in this prediction. But more considerably more valid estimates of future trajectories can be gained by knowledge of intent of one or both aircraft in a conflict. Such intense inferencing can be gained from a variety of sources: current velocity vectors, filed flight plans, information resident in the flight management system, even the active queries of the pilots involved. The further into the future that reliable intent inferences can be made, the more flexibility the pilots will have in selecting routes to avoid conflict situations.

Air Traffic Control

All players acknowledge the critical sustaining role of air traffic control in a free flight management system. The role is seen in at least two ways:

1. Any free flight system will need to include both unconstrained (free flight) and constrained airspace. In the latter, conditions of high traffic density or the need to maintain regular flow militate against user-preferred routine.
2. There is always a danger that a potential conflict situation may develop for which pilots involved are unable or unwilling to formulate a satisfactory solution. Air traffic control then must be alert to "bail out" the pilots from catastrophe in such situation.

A large number of issues must be addressed and resolved before determining if free flight system is feasible in an airspace whose regulators and occupants are committed to safety as a primary goal. We discuss these issues below in two categories, those pertaining to the airspace system as a whole, and those focusing more directly on human factors.

2.2 System Level Issues

Air Traffic Control Role

The role of air traffic control in free flight regime will undoubtedly be a critical and controversial issue. Indeed, one of the thornier issues concerns the appropriate level of authority that should be maintained by air traffic control. On one extreme is a system in which aircraft maneuver as they choose, allowing air traffic control to be only a passive monitor of the changing trajectories, until or unless these lead to danger, and then intervening with control. A more conservative system will require pilots to inform air traffic control of their maneuvers but proceed to carry them out unless vetoed by air traffic control. Still more conservative is a system not unlike that in existence today, in which pilots request deviations and air traffic control approves. However under free flight regime, such requests would be far more frequent, given the pilots would have the equipment (e.g. cockpit display of traffic information) and training to carry them out safely.

Pilot's and the Airline Operations Center's Roles

We are assuming through the whole discussions that the pilot is the one calling the shots in a free flight regime. However from the standpoint of commercial aviation the pilot is not necessarily the best originator of unconstrained maneuver plans. Instead the airline operations center and its agent the aircraft dispatcher, will probably have far better global knowledge of weather patterns, winds, traffic scheduling, and regional traffic density, in order to make more nearly optimal decisions on route and trajectory changes. Hence although the pilot may become free from air traffic control constraints, these may be replaced by constraints from the dispatcher.

System-Wide Efficiency

On paper, convincing cases can be made for the cost savings of direct routings and other free flight concepts. However in practice savings that appear in one place may be lost in others. It is revealed that free flight can considerably lessen the cruise flight time en route between TRACONS (constrained airspaces). But much of the time saved may then be lost, as a large stack of rapidly arriving aircraft must now wait at the feeder gate to a TRACON, in order to be handled in a less efficient, more sequential fashion by air traffic control. Also losses of efficiency may emerge from a group behavior in ways that cannot easily be predicted in advance. Assume a phenomenon whereby several aircraft, all requesting the same preferred routing, created a bunching on that preferred route that ultimately slowed their flight, and in some cases required redirection back to the earlier nonpreferred route, now with a considerable loss of time. In this case flight time is not saved, nor is any workload reduced for the controller. It may well be difficult or impossible to predict other such system-wide effects until or unless full operational test of the system is in place.

Safety versus Efficiency

The pressure towards free-flight is primarily efficiency driven. The FAA has rightfully maintained a conservative stance, driven by safety, in responding to pressures to move towards free-flight. But given the recent commitment to reduce accident rates, it can be argued that any radical changes to an already safe system will at least have the possibility of being safety-compromising. The goal is to achieve developing sophisticated modeling of both safety and efficiency parameters of new technologies and procedures. Valid airspace safety models that include contributions of human operator (pilot or controller) processing are greatly needed in order to predict safety implications of free-flight, and compare these implications with those supported by higher levels of ground based automation, discussed below.

Equipment

Free flight demands special technical equipment: accurate global positioning systems, automatic dependent surveillance communications, and high-resolution cockpit displays of traffic information. Using such technology the position of a fully equipped commercial aircraft can be estimated within a standard deviation horizontally and vertically. However any airspace that contains at least one aircraft without such equipment is placed at risk in a free flight regime.

2.3 Human Factor Issues

Many of the human factor issues to be addressed in free flight pertain to the infrequent situations in which two aircraft have selected routes that will bring them into conflict.

Level of Air Traffic Control Authority

How easy will it be for air traffic control to veto inappropriate maneuvers and flight plans, or should these indeed be subject to preapproval? If a controller's conflict probe enables him to predict a conflict within 20 minutes, should the controller intervene or offer an advisory to two aircraft in free flight? One issue concerns the extent to which controllers, rather than pilots, may have better skills, and more global displays, to appreciate global traffic patterns and may therefore be better equipped than pilots to judge the long-range implications of maneuvers.

Equally important are issues associated with ambiguity in authority. Almost any free flight system assumes regions (or times) in which air traffic control has authority and those in which they do not. At issue are the transition periods between such authority assignments (e.g. transferring from unconstrained to constrained airspace or from pilot-centered strategic maneuvering to resolve a conflict.) Such regions invite ambiguity in turn will invite noncooperative maneuvering or unnecessary and time-consuming negotiations.

Situation Awareness

The controller's awareness of the big picture may be degraded under free flight for one of three reasons. First, psychological research has shown that when people do not actively direct changes but only observe them passively, they are less likely to remember them. Hence a controller who passively witnesses a pilot changing altitude will be less likely to be aware of and remember the implications of that new altitude for another aircraft, than if the controller had actively selected the change (or even had to consider and approve it). Second, an airspace that functions under free flight rules will, almost by definition, lose the structured order that enables the controller to easily grasp the big picture. Aircraft will no longer be flying linearly along predefined routes, and flight levels may no longer be evenly spaced and predictably occupied. It is quite possible that an airspace under free flight

yield unpredictable shifts in traffic density, and this in turn may require some degree of “dynamic resectorization”. Given the strong dependence of the controller’s mental model on the static and the above characteristics produce greater difficulty in maintaining situation awareness. Finally, free flight separation algorithms, are likely to be time based rather than space based. Space can be easily visualized by the controller, but time less so. It is unclear the extent to which this shift may also inhibit controller situation awareness.

Controller Workload

Workload and situation awareness are closely related, and the mediation of these two concepts by a free flight regime leads to several possible implications. Under routine conditions, controller workload in decision making and communications may be reduced by free flight. The likely decrease in traffic complexity will impose greater cognitive workload in trying to predict traffic behavior to maintain adequate situation awareness. Furthermore, controller workload is likely to be substantially increased under the infrequent circumstances in which two or more aircrafts cannot negotiate a non-conflict solution and the controller must intervene. Also, increased efficiency of free flight in the unconstrained regions may produce traffic bottlenecks at the borders of the constrained regions, hence imposing high workload to deal with the resulting traffic rush.

Negotiations

A minimum of two players are potentially involved in any conflict resolution scenario. If conflicts are predicted far in advance then only the two pilots may be involved in negotiations to avoid. If such negotiations are not initiated or not completed progressively close to the predicted time of the conflict, air traffic control is more likely to get involved and possibly desire to intervene. It is also easy to imagine circumstances in which a third aircraft can be a party to the negotiations if a maneuver by one of the first two turn it toward the third. The application of negotiation theory to the free flight regime will become simpler to the extent that clearly defined rules and instructions are laid out.

Chapter 3

Proposed Approach

3.1 Framework

We consider the concept of resolving conflicts arising among aircrafts following a cooperative approach , i.e. all aircraft involved in a conflict collaorat to its resolution. This approach is based on the following central assumptions:

a.Aircraft are assumed to cruise within a fixed altitude layer (the layer structure is the same as the one described in [2].) Aircraft can thus be modelled in a purely kinematic way, as points in a plane with an associated fore axis, that indicates the direction of motion, and conflict envelope radius. The task of each vehicle is to reach a given goal configuration (start and goal configurations may represent waypoints planned for the aircraft by the higher level planner).

b.All interacting aircraft cooperate towards optimization of a common goal, as agents in the same team. The common goal is to reach the final configuration avoiding all possible conflicts. This applies to all aircraft in the same airspace, defined as a zone in which they can exchange information on positions, velocities and goals.

c.In ATC literature [2-10], two cases have been considered for conflict resolution problems: in the first case we study aircraft maneuvers consisting of instantaneous velocity changes and in the second case heading angle changes are allowed. Our preposition consists a combined approach that allows aircrafts simultaneous changes in both velocity an heading angle.

The problem of finding the shortest conflict-free paths in the combined approach can be modeled as a non-linear Mixed-Integer Programming (MIP) which may be solved using optimization tools such a GAMS[11].

3.2 Problem Statement

We consider two aircraft sharing a confined airspace. Each aircraft is an autonomous vehicle that flies on a horizontal plane. Furthermore, each aircraft has an initial and a final, desired configuration (position, heading angle) and the same goal is to reach the final configuration in minimum time while avoiding conflicts with other aircraft. A conflict between two aircraft occurs if the aircraft are closer than a given distance d (current enroute air traffic control rules often consider this distance to be 5 nautical miles).

Aircrafts are identified by points in the plane (position) and angles (heading angle, direction) and thus by a point $(x,y,\theta) \in \mathbb{R} \times \mathbb{R} \times \mathbb{S}^1$. Let $(x_i(t), y_i(t), \theta_i(t))$ be the configuration of aircraft i at time t , and $(x_j(t), y_j(t), \theta_j(t))$ the configuration of aircraft j at the same time t . A conflict occurs among these aircrafts when the distance between them is less than d i.e. a conflict between aircraft i and j occurs if for some value of t ,

$$\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2} < d. \quad (3.1)$$

Considering the aircraft as disks of radius $d/2$, the condition of non-conflict between aircraft is equivalent to the condition of non intersection of the discs. In the following we refer to those as the *safety disc* of the aircraft. The following sections will detail the construction of non-linear avoidance constrains that are equivalent to (3.1).

As mentioned previously, to avoid possible conflicts we consider two different cases:

- a.** we allow aircraft to change the velocity of flight but the direction of motion remains fixed. We will refer to this case as the *Velocity Change Problem* (VC problem).
- b.** aircraft fly at the same velocity u and are only allowed to change instataneously the direction of flight. We will refer to this case as the *Heading Angle Problem* (HAC problem).

In both cases each aircraft is allowed to make maneuver, at time $t = 0$, to avoid all possible conflicts with other aircraft. We assume that no conflict occurs at time $t = 0$.

Both problems considered can be formulated as linear mixed integer optimization problems that can be solved using several optimization tools. However, in real life we can not expect an aircraft to be able to make only a specific type of maneuver (velocity change or heading angle change) in order to avoid possible conflicts. Instead, we propose a combined approach of the previous methods that allows aircrafts to change simultaneously both their velocities and heading angles and as a result, avoiding possible conflicts with less overall deviations from their nominal paths. Due to this approach, the resulting problem will be non-linear, therefore more computationally expensive than before, but much more realistic and efficient in real life situations. As for the computational cost is concerned, it turns out that it is limited to a few seconds more than the cost in the previous approaches.

We can now define as q_i the velocity change and as p_i the heading angle change of the i -th aircraft. The problem consists in finding a velocity change and a heading angle change for each aircraft, in order to avoid possible conflicts while deviating as little as possible from the original flight plan. In our approach where we are dealing with a possible conflict between two aircrafts, this deviation is formulated as an objective function which we will try to minimize. The problem can be formulated as non-linear optimization problem with non-linear constraints and some boolean variables.

3.3 Problem Formulation

We will now refer to the VC and HAC problem in the concept of producing the conflict avoidance constrains for both of them and then proceed to the combinatory approach.

3.3.1 The VC problem

The VC problem consists of aircraft that fly along a given fixed direction and can maneuver only once with a velocity variation. We consider two aircrafts denoted 1 and 2 respectively. The i -th ($i = 1,2$) aircraft changes its velocity of a quantity q_i that can be positive (acceleration), negative (deceleration), or null (no velocity change). Each aircraft has upper and lower bounds on the velocity u_i :

$$v_{i,min} \leq v_i \leq v_{i,max}. \quad (3.2)$$

As a matter of fact, for commercial flights, during enroute flight, we usually have:

$$\frac{v_{i,max} - v_{i,min}}{v_{i,min}} \leq 0.1. \quad (3.3)$$

The problem then is to find an admissible value of q_i ,for each aircraft, such that all conflicts are avoided and such that new velocity satisfies the upper and lower bounds. Hence given the initial velocity u_i after a velocity variation of amount q_i the following inequalities must be satisfied:

$$v_{i,min} \leq v_i + q_i \leq v_{i,max}. \quad (3.4)$$

Let (x_i, y_i, θ_i) , $i = 1, 2$, be the aircraft position and direction of motion and u_i be the initial velocity. Referring to Figure 3.1, we consider the two velocity vectors:

$$\hat{v}_1 = \begin{bmatrix} (v_1 + q_1)\cos\theta_1 \\ (v_1 + q_1)\sin\theta_1 \end{bmatrix}; \quad (3.5)$$

$$\hat{v}_2 = \begin{bmatrix} (v_2 + q_2)\cos\theta_2 \\ (v_2 + q_2)\sin\theta_2 \end{bmatrix}; \quad (3.6)$$

and the difference vector:

$$\hat{v}_1 - \hat{v}_2 = \begin{bmatrix} (v_1 + q_1)\cos\theta_1 - (v_2 + q_2)\cos\theta_2 \\ (v_1 + q_1)\sin\theta_1 - (v_2 + q_2)\sin\theta_2 \end{bmatrix}; \quad (3.7)$$

The two lines parallel to $\hat{u}_1 - \hat{u}_2$ that are tangent to aircraft 2, localize the segment on the direction on motion of aircraft 1 (refer to Figure 3.1). We will refer to this segment as the *shadow* of aircraft 2 along the direction of 1. A conflict occurs if the aircraft 1 with his safe disc intersects the shadow generated by aircraft 2, or vice-versa since $\hat{u}_1 - \hat{u}_2$ and $\hat{u}_2 - \hat{u}_1$ are parallel.

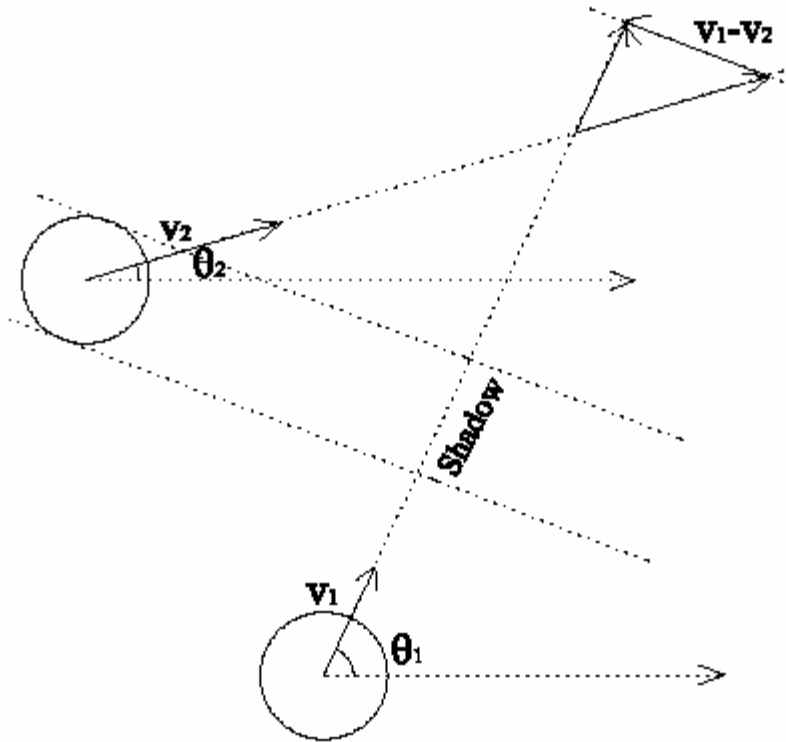


Figure 3.1. Geometric construction for conflict avoidance constraints in the case of intersecting trajectories for the VC problem. In this case aircraft 1 do not intersect the shadow generated by aircraft 2 then no conflict will occur between the two aircraft.

Consider now the two non-parallel straight lines that are tangent to the discs of both aircraft (see Figure 3.2). Let l_{12} , r_{12} be the angles between these two straight lines and the horizontal axis. We have $l_{12} = \omega_{12} + \alpha$ and $r_{12} = \omega_{12} - \alpha$ with $\alpha = \arcsin(d / A_{12})$ where A_{12} is the distance between the two aircraft and ω_{12} is the angle between the line that joins the aircraft and the x -axis.

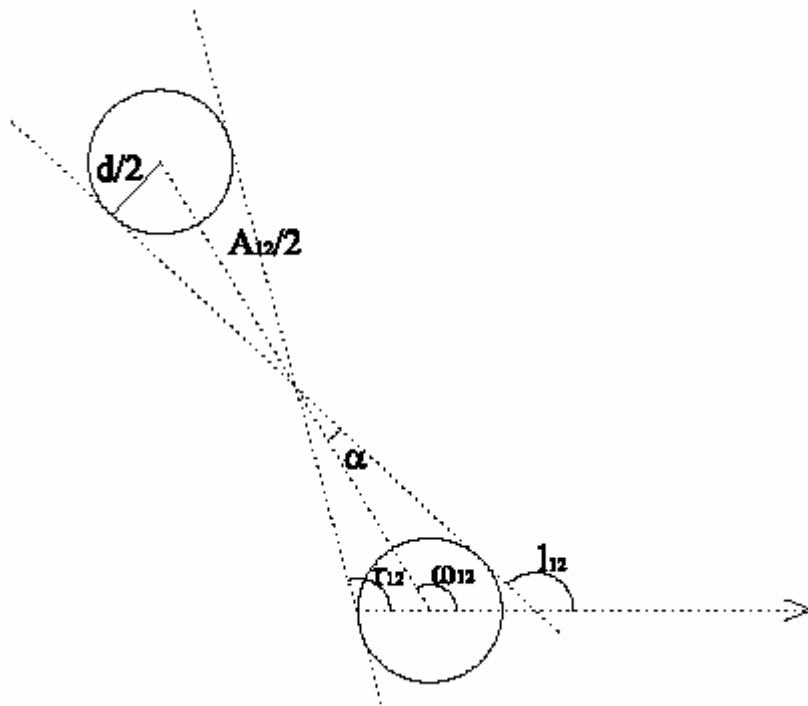


Figure 3.2. The two non parallel straight lines tangent to the safety discs of radius $d/2$ for two aircraft at distance $A_{12} / 2$.

From the above we can say that no conflict occurs if:

$$\frac{(v_1 + q_1)\sin\theta_1 - (v_2 + q_2)\sin\theta_2}{(v_1 + q_1)\cos\theta_1 - (v_2 + q_2)\cos\theta_2} \geq \tan(l_{12}) \quad (3.8)$$

or

$$\frac{(v_1 + q_1)\sin\theta_1 - (v_2 + q_2)\sin\theta_2}{(v_1 + q_1)\cos\theta_1 - (v_2 + q_2)\cos\theta_2} \leq \tan(r_{12}) \quad (3.9)$$

Now we have the or-constraints for the VC problem. We also notice that the above constraints are linear in the velocity variation q_i , $i = 1, 2$ and so are the constraints for the upper and lower bounds in (3.4).

Obviously, the VC method can not produce a solution in a possible head to head conflict because any velocity variation would not make any difference. A solution to this problem can be produced with the HAC method which we are going to analyze.

3.2 The HAC problem

The HAC problem consists of aircraft that fly at the same constant velocity u and that can maneuver only once with an instantaneous heading angle deviation. We consider two aircraft denoted 1 and 2 respectively. The i -th aircraft ($i = 1, 2$) changes its heading angle of a quantity p_i that can be positive (left turn), negative (right turn), or null (no deviation).

The problem then is to find an admissible value of p_i for each aircraft such that possible conflicts are avoided with the new heading angle (direction of flight), $\theta_i + p_i$. We are now going to formulate the non-conflict constraints for the HAC problem as inequalities that are linear in the unknowns p_i and that are function of the aircraft's initial configurations (x_i, y_i, θ_i) , $i = 1, 2$. Let $(x_i, y_i, \theta_i + p_i)$ be the aircraft's states after the maneuver of amplitude p_i . We will show that it is possible to predict the existence of conflicts between the two aircraft based on those aircraft's initial configurations. The constraints will be obtained by geometrical construction.

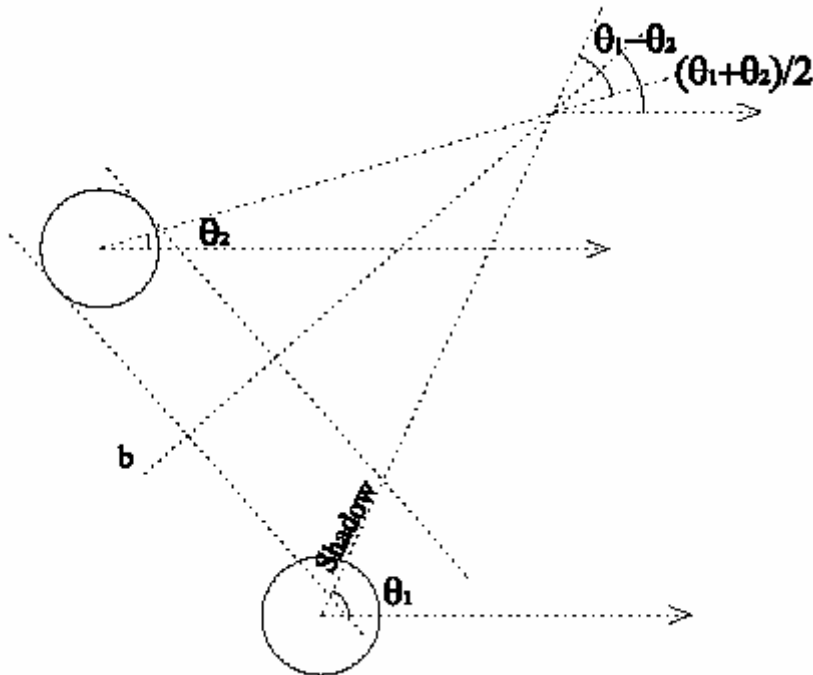


Figure 3.3. Geometric construction for conflict avoidance constraints in the case of intersecting trajectories for the HAC problem. In this case aircraft 1 intersects the shadow of aircraft 2, then a future conflict has been detected.

Referring to Figure 3.3, consider the two aircraft (x_1, y_1) and (x_2, y_2) with heading angles θ_1 and θ_2 respectively. Momentarily consider $p_1 = p_2 = 0$ for simplicity reasons. Consider the angle of amplitude $(\theta_1 - \theta_2)$ comprised within the aircraft flight directions. The bisector b is then a straight line that forms an angle $(\theta_1 + \theta_2) / 2$ with the x-axis, while the orthogonal to the bisector forms an angle of $m_{12} = (\theta_1 + \theta_2 + \pi) / 2$ with the x-axis.

The family of straight lines of slope $\tan(m_{12})$, orthogonal to the bisector, represents also the projection of one aircraft along the direction of motion of the other. The two lines in this family that are tangent to aircraft 2 localize a segment on the direction of motion of 1 (refer to Figure 3.3): we will refer to this segment as the *shadow* of aircraft 2 along the direction of 1. As described in the previous section, a conflict occurs if aircraft 1 with his safe disc intersects the shadow generated by aircraft 2, or vice-versa since the angle m_{12} is symmetric in θ_1 and θ_2 .

Again consider Figure 3.2, let l_{12} , r_{12} be the angles between these two straight lines and the horizontal axis. We have $l_{12} = \omega_{12} + \alpha$ and $r_{12} = \omega_{12} - \alpha$ with $\alpha = \arcsin(d / A_{12})$ where A_{12} is the distance between the two aircraft and ω_{12} is the angle between the line that joins the aircraft and the x-axis. The condition of non intersection of the shadows is equivalent to the following condition:

$$m_{12} \leq r_{12} \tag{3.10}$$

or

$$m_{12} \geq l_{12} \tag{3.11}$$

where $m_{12} = (\theta_1 + \theta_2 + \pi) / 2$

These *or-constrains* will also be included in our model. The model of the HAC problem is now complete. In the case of heading angle maneuvers we can consider in the model also other kind of constrains. For example we can consider the possible existence of forbidden zones of airspace due to severe weather or overloaded airspace, see Figure 3.4. To model those forbidden zone, it is sufficient to consider bounds of the heading angle deviations.

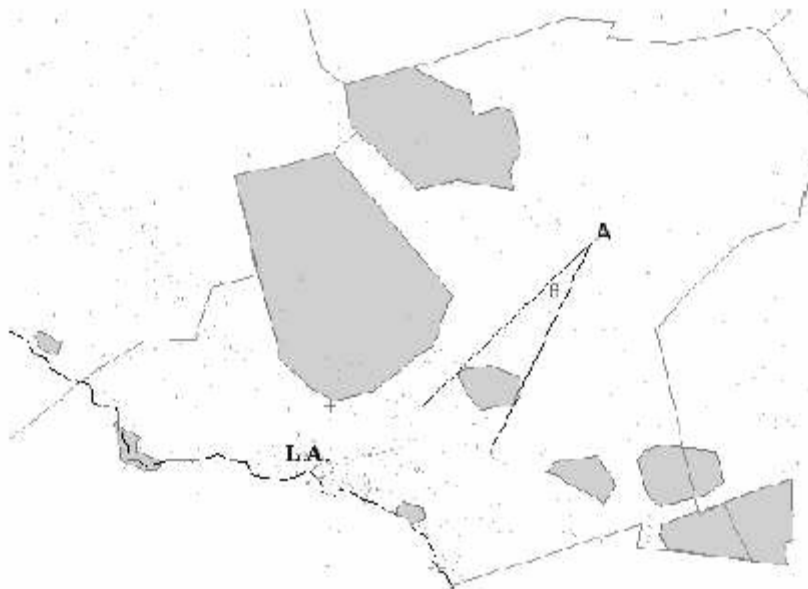


Figure 3.4. Example of forbidden sectors in the Los Angeles control sector. For the aircraft *A* we need to introduce more constrains on the direction of flight due to forbidden zones of airspace.

3.4 The Combined Approach

As we mentioned earlier, our proposed approach involves a combined approach of the two previously analyzed methods (VC and HAC). We will proceed in defining the non-conflict constraints for the combined approach and also in defining the cost function we will try to minimize.

3.4.1 The Final Set of Conflict Constraints for the Combined Approach.

We proceed with including in our model the conflict constraints from the VC problem (3.8,3.9) with a little difference. The difference is that we replace in these equations the heading angle θ_i which is constant in the VC problem, with the heading angle plus the heading angle deviation, thus $\theta_i + p_i$ where p_i is the angle deviation of the i -th $i=1,2$ aircraft from his originally scheduled flight plan. The final set of constraints for the two aircraft we have discussed is used for simulation methods and is:

$$\frac{(u_1 + q_1)\sin(\theta_1 + p_1) - (u_2 + q_2)\sin(\theta_2 + p_2)}{(u_1 + q_1)\cos(\theta_1 + p_1) - (u_2 + q_2)\cos(\theta_2 + p_2)} \geq \tan(l_{12}) \quad (3.12)$$

or

$$\frac{(u_1 + q_1)\sin(\theta_1 + p_1) - (u_2 + q_2)\sin(\theta_2 + p_2)}{(u_1 + q_1)\cos(\theta_1 + p_1) - (u_2 + q_2)\cos(\theta_2 + p_2)} \leq \tan(r_{12}) \quad (3.13)$$

where l_{12} and r_{12} are the same angles which are referred in Figure 3.2, for the VC problem. We will also include in our model the constraints of the HAC problem (3.10,3.11). In this case, we don't have to modify them by replacing the constant velocity with a variant one because the velocity doesn't appear in the equations. So, we will use the equations (3.10,3.11) as described above in the HAC problem analysis.

A very important issue now needs to be addressed. We have come to construct the final set of constraints for our combined approach of the problem and now we have to define the metric we are going to optimize, the cost function of our method. Gazing through the ATC literature[2-10], we considered that one suitable function would be that of the sum of the absolute prices of both aircraft's velocity and heading angle deviations. More specifically this function would be of the form:

$$z_1 = |p_1| + |p_2| + |q_1| + |q_2| \quad (3.14)$$

The sum of the **absolute** prices is very much different than the sum of the prices of the deviations, because every deviation can be negative, positive or null, as mentioned in the above sections. Furthermore, a choice of the above cost function would be of the general idea that the smaller the velocity and heading angle deviations would be, the smaller would be the overall deviation from the aircraft's scheduled route and the correction of the aircraft's path, after avoiding the possible conflict, too. We have also considered one more cost function which we will use for simulation purposes and that function is of the form:

$$z_2 = p_1^2 + p_2^2 + q_1^2 + q_2^2 \quad (3.15)$$

We came to this second decision of cost function, in order to take advantage of the behaviour of the function $y=x^2$, which for small values of x produces small values of y and in contrast for big values of x , y becomes quadratically larger than the y produced in the function $y=|x|$. With this property, we are giving our optimization software a further assistance towards optimization.

In conclusion, we note that our problem as it has been transformed is now a non-linear problem, soon to be transformed in a mixed-integer non-linear problem in the later section. As a logical result we will have to expect larger executional times and slower performance than before, which may have an impact in real time situations, but we have to consider that our approach is much more realistic than before since we allow both velocity and heading angle deviations. In the next section, we are going to address the issue of the formulation of the above mentioned constraints, in a suitable manner so that they can be used as an input in an optimization software of our choice, in order to produce the desired results.

3.4.2 Formulation of the final conflict constains into mixed-integer programming constrains.

We are now faced with a non-linear optimization problem awaiting to be solved with efficiency and a relative speed in computational time. For that task, many software optimization packages are provided such as CPLEX, DICOPT, SBB etc . Our choice was the GAMS software optimization package (www.gams.de) which is at this point the leading tool for the development, solution and management of large scale optimization problems. It provides a high-level language for the compact representation of large and complex models and furthermore it allows unambiguous statements of algebraic statements, thus it provides a very friendly interface. For further information about the basic features of GAMS, the reader is encouraged to look in Appendix A.

We will now examine how the final conflict constains should be formulated in order to be put as input in GAMS. As any optimization package, GAMS requires that the constrains used must be of the form of *and-constrains* which means that they have to be satisfied simultaneously. To be more specific, it can solve optimization problems of the form:

$$\min f(x) \tag{3.16}$$

such that

$$g(x) \leq 0 \tag{3.17}$$

where $f(x)$ is a function of n real variables $x=(x_1, x_2, \dots, x_n) \in \mathcal{R}$ and is subject to a set of inequality constrains. This means that the constrains $g_i(x)$ must all be valid at the same time (g_1 AND g_2 AND ... AND g_i). As we have mentioned in the previous sections our constrains are of the form of *or-constrains* and so we have to formulate them once more. This conversion can be done by introducing some boolean variables. Below we present a comprehensive example of how we can convert *or-constrains* into *and-constrains*.

Let's consider an example group of constrains:

$$\begin{aligned} c_1 &\leq 0 \\ &\text{and} \\ c_2 &\leq 0 \end{aligned} \tag{3.18}$$

or

$$\begin{aligned} c_3 &\leq 0 \\ &\text{and} \\ c_4 &\leq 0 \\ &\text{and} \\ c_5 &\leq 0 \end{aligned} \tag{3.19}$$

or

$$\begin{aligned} c_6 &\leq 0 \\ &\text{and} \\ c_7 &\leq 0 \end{aligned} \tag{3.20}$$

where $c_i, i=1\dots7$ are linear or non-linear expressions in the decision variables (which in our case are the velocity and heading angle deviation).

The way to transform these *or-constrains* into more convenient *and-constrains* is to introduce Boolean variables[12]. Let f_k with $k = 1,2,3$ be a binary number that takes value zero when one of the *or-constrains* is active and 1 otherwise (for example $f_1 = 0$ if constrains c_1 and c_2 are active, $f_1 = 1$ otherwise). Let M be a large arbitrary number, then the previous set of constrains is equivalent to:

$$\begin{aligned}
c_1 - Mf_1 &\leq 0 \\
c_2 - Mf_1 &\leq 0 \\
c_3 - Mf_2 &\leq 0 \\
c_4 - Mf_2 &\leq 0 \\
c_5 - Mf_2 &\leq 0 \\
c_6 - Mf_3 &\leq 0 \\
c_7 - Mf_3 &\leq 0 \\
f_1 + f_2 + f_3 &\leq 2
\end{aligned} \tag{3.21}$$

The above constrains are now *and-constrains* and so we came to overcome the previous difficulty. It is clear now tha we are looking at the so called Mixed-Integer Programming (MIP) problem because we have two kinds of variables: normal variables that can take any value, and binary variables that can take only 0 and 1. It can be easily understood that MIP problems are much more complex that Pure-Integer Programming problems where the decision variables can only take binary variables.

Finally, we are now able to transform our final conflict constrains from *or-constrains* into *and-constrains* and use them as input in GAMS. In the next section we will present the results produced by the optimization package for various topology scenarios, and comment about their characteristics.

3.5 Simulation and case scenarios

The topology we choose for simulation purposes, is shown at Figure 3.5. The basic idea is that we have a circle of radius R , in which we scan for possible conflicts. We will refer to this circle as the *control circle* from now on. Aircraft that are going through the circle have an initial and final destination. The initial configuration of an aircraft consists of the velocity and heading angle at the point of entry into the circle and the final configuration is the velocity and heading angle at the point that lies at the original aircraft direction at distance $1.5d_i$ where d_i is the length of i -th aircraft's trajectory in the circle.

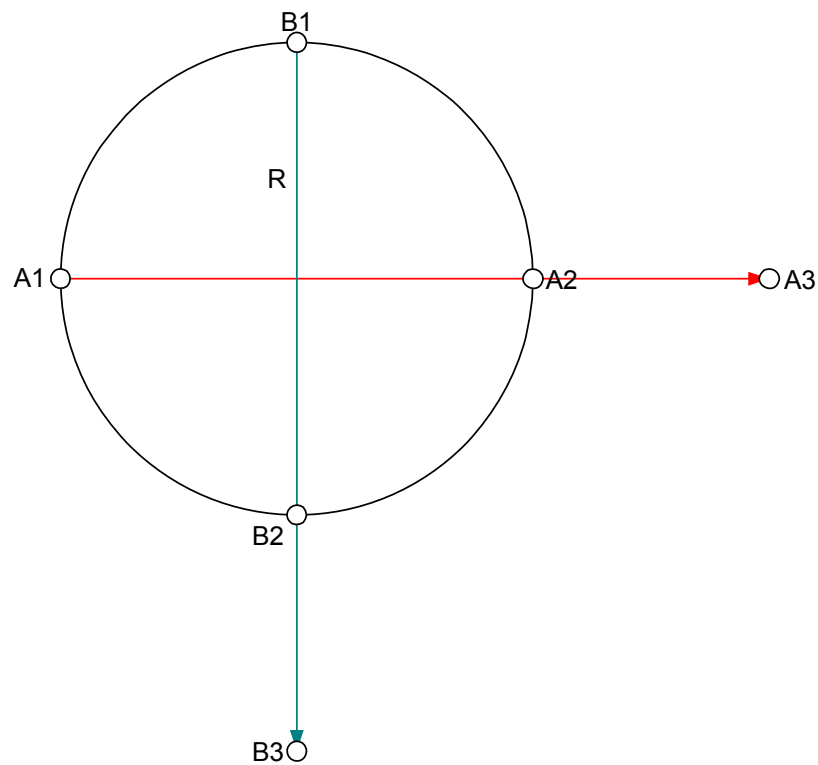


Figure 3.5. Initial and final configuration points of two aircrafts traversing the control circle.

For example in Figure 3.5 the initial configurations of aircraft A,B are A_1, B_1 and the final configuration points are A_3, B_3 respectively, while $A_1A_3=1.5A_1A_2$ and $B_1B_3=1.5B_1B_2$. Any aircraft that enters the control circle checks for possible conflicts with any other aircraft that might be in the circle. When a conflict is detected, the aircrafts will do the appropriate maneuvers produced by the solution of mixed-integer NLP, and will continue travelling with the new velocity and heading angle, until they reach the point of exit from the control circle. When they finally reach the point of exit from the control circle, they both will do a corrective change in velocity and heading angle in order to reach their original final configuration point. In Figure 3.6 a comprehensive example is presented :

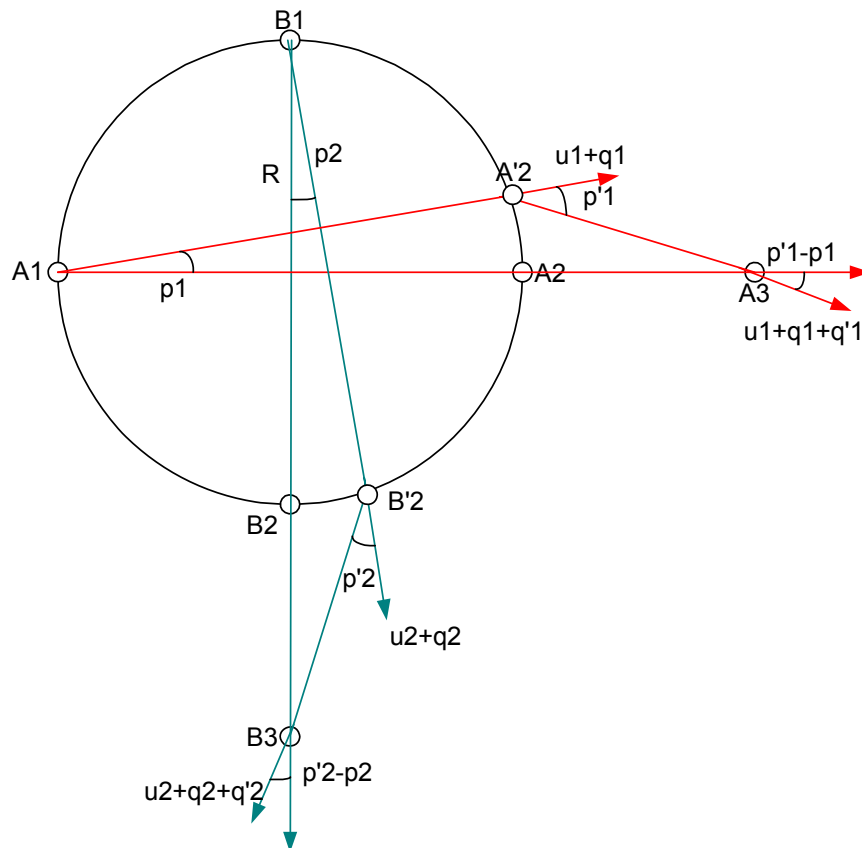


Figure 3.6. A comprehensive example figure about the aircraft's behaviour before and after conflict resolution.

Both aircrafts A and B enter the control circle with the same velocity ($u_1 = u_2$) and follow the trajectories shown in Fig. 3.6 with initial heading angles $\theta_1=0$ and $\theta_2=-\pi/2$ for A,B respectively. It is obvious that if things stay the same, a conflict will occur in the center of the circle. At this point, both aircrafts make instantaneous changes in their velocities and heading angles (changes which are decided by the optimization tool) in order to avoid conflict. At this example we assume that the aircrafts change both their velocities and heading angles. This is not necessarily happening in all the different cases of possible conflict. In another case scenario, it may be found that an optimal maneuver for one aircraft might be zero velocity change and some heading angle deviation, and the other way around. In this cases the optimization software just assigns zero to the attribute that doesn't need to be changed.

Let us return in analyzing what we see in Fig. 3.6. After conflict resolution, aircrafts A and B travel with velocities $(u_1 + q_1)$ and $(u_2 + q_2)$ and heading angles p_1 and $(p_2 - \pi/2)$ respectively. These attributes now determine their trajectories until they exit the control circle from the points A'2 and B'2 respectively. New changes are now been made, corrective changes so that both aircraft return to their predetermined nominal paths. Aircraft A and B change their velocity by q'_1 and q'_2 and also turn by a magnitude of p'_1 and p'_2 respectively, and so they finally reach their originally scheduled destination A3 and B3 respectively. Once they are at their final configuration points, they both have to make another deviation in the heading angle by a magnitude of (p'_1-p_1) and (p'_2-p_2) respectively so that they completely resume their originally scheduled nominal path.

Let's now proceed in discussing various case scenarios, and produce results with the simulations done with the help of the optimization tool. These case scenarios involve two aircrafts resolving a conflict. Both aircraft enter the control circle of radius $108km/min$ at the same time, and with the same velocity $15 km/min$, which is actually the maximum speed allowed by restrictions for passenger safety&comfort and fuel consumption. The minimum safe distance between the two aircraft has been set to $9km$. The aircrafts' initial trajectories, pass through the center of the circle and given the above mentioned characteristics (about entering at the same time with the same velocity), we are ensured that a conflict is about to occur in the center of the control circle. For each scenario we have simulated, we will present you two figures produced in the VISIO 2000 environment. The first one shows the aircrafts' configuration points and trajectories if no maneuvers are made to avoid conflict. The second one shows the aircrafts' trajectories, after the resolution of the conflict, produced with the data acquired from simulations in GAMS.

The final configuration points of the aircraft are not presented in these figures, because our particular interest is to observe the motion of the aircraft in the control circle, when resolving a conflict. Of course, in these figures one will be able to see the heading angle deviations of the aircraft and not the velocity deviations, and for that reason each case scenario is followed by a table showing velocities and heading angles, before and after conflict resolution, so we can see the produced results numerically as well. We can now proceed in presenting 8 simulated case scenarios, for each of z_1, z_2 which are, as we previously said, the objective functions which we will try to minimize:

3.5.1 Case Scenario 1.1

$$z = z_1$$

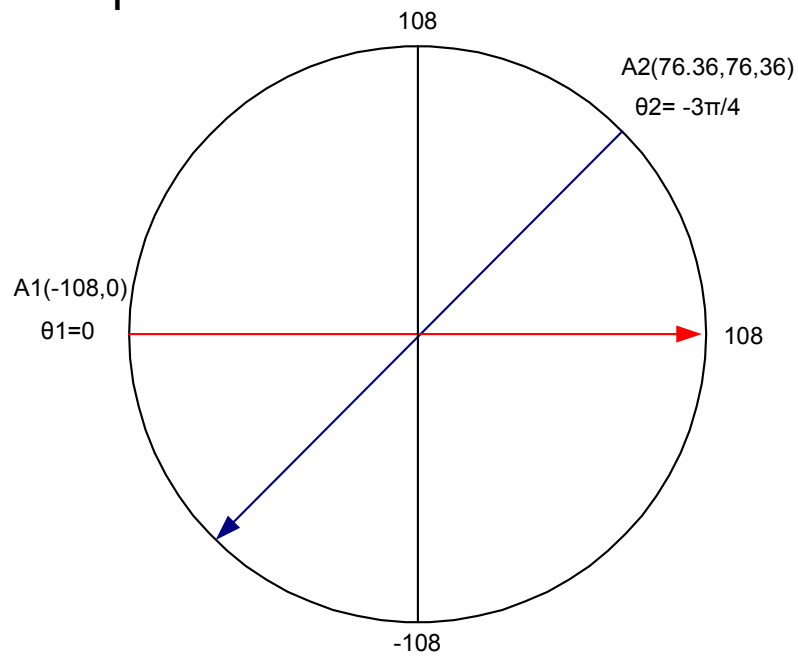


Figure 3.7.Scenario 1.1's distributed aircraft and their trajectories, before conflict resolution.(z=z1)

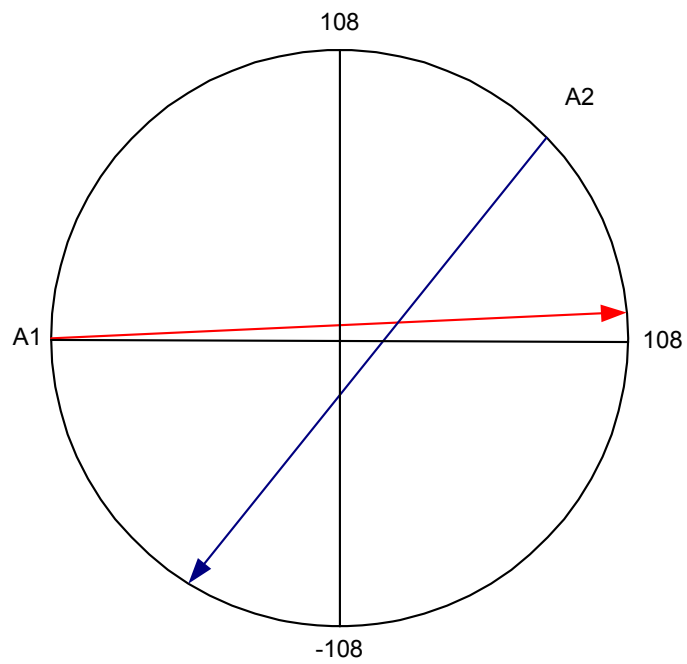


Figure 3.8.Scenario 1.1's distributed aircraft and their trajectories, after conflict resolution.(z=z1)

Scenario 1.1	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	-0.187	0.123
Heading angle before CR(rad)	0	-2.355
Heading angle change after CR (rad)	0.011	0.08

Figure 3.9. Table showing the velocities, heading angles and overall deviations, for scenario 1.1's distributed aircrafts. CR stands for Conflict Resolution.

3.5.2 Case Scenario 1.2

$z = z_1$

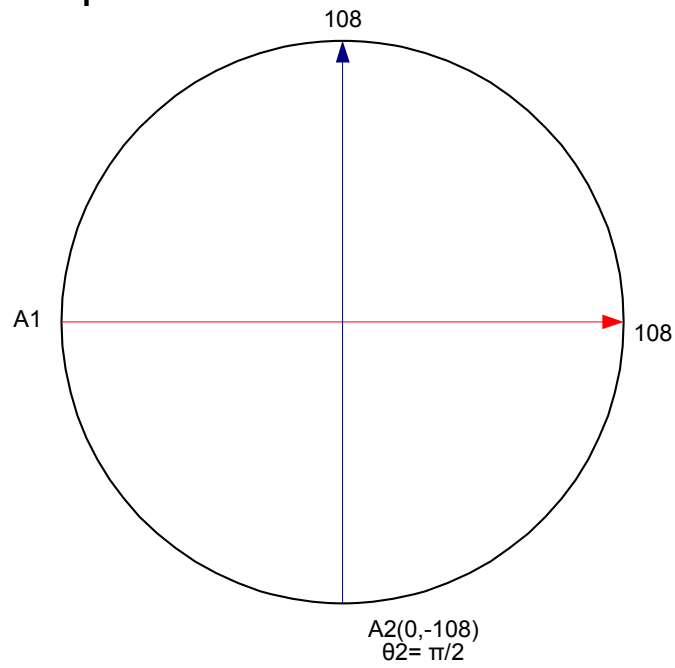


Figure 3.10. Scenario 1.2's distributed aircrafts and their trajectories, before conflict resolution. ($z=z_1$)

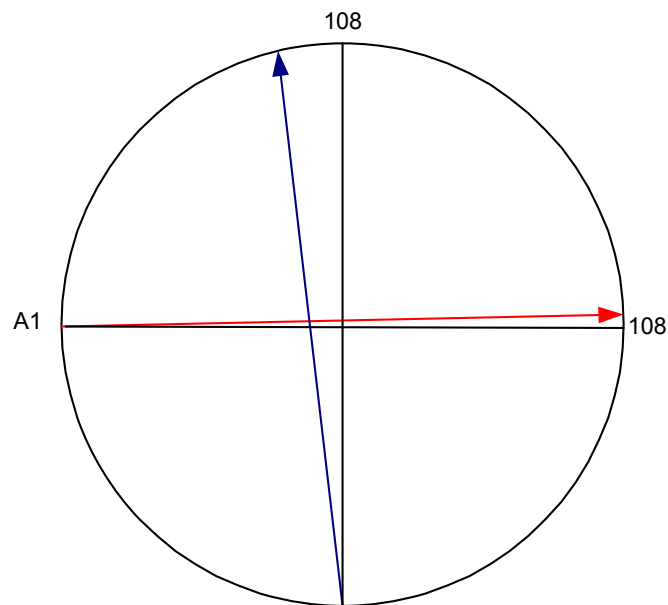


Figure 3.11. Scenario 1.2's distributed aircrafts and their trajectories, after conflict resolution. ($z=z_1$)

Scenario 1.2	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	0.746	-0.832
Heading angle before CR (rad)	0	1.57
Heading angle change after CR (rad)	0.007	0.111

Figure 3.12. Table showing velocities, heading angles and overall deviations, for scenario 1.2's distributed aircrafts. CR stands for Conflict Resolution.

3.5.3 Case Scenario 1.3

head to head

$z = z_1$

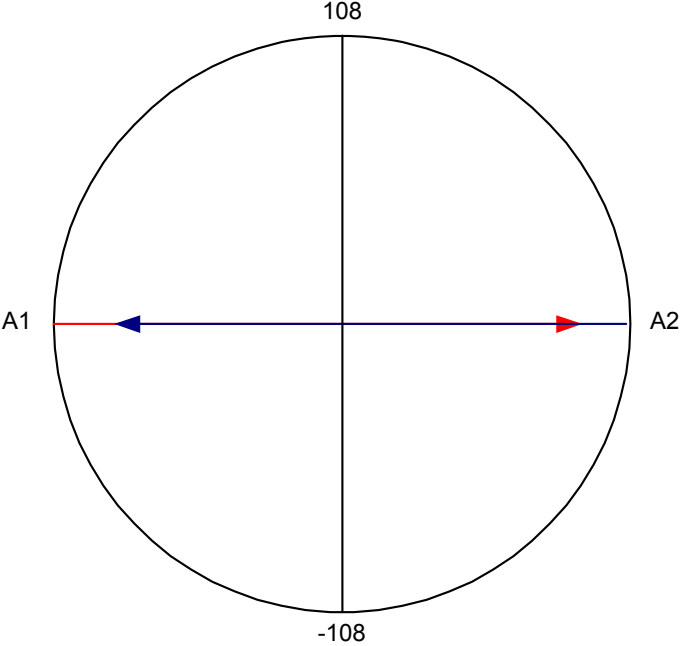


Figure 3.13.Scenario 1.3's distributed aircraft and their trajectories, before conflict resolution.($z=z_1$)

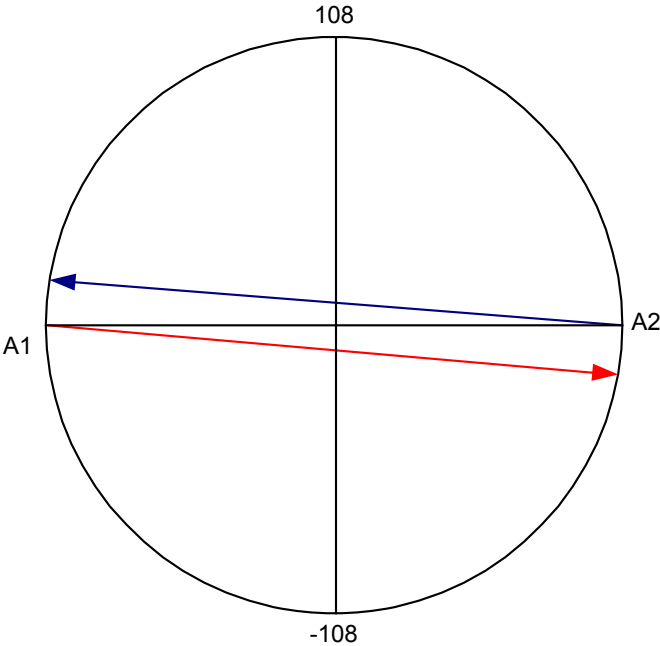


Figure 3.14.Scenario 1.3's distributed aircraft and their trajectories, after conflict resolution.($z=z_1$)

Scenario 1.3	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	0.240	-0.114
Heading angle before CR (rad)	0	3.14
Heading angle change after CR (rad)	-0.041	-0.042

Figure 3.15. Table showing velocities, heading angles and overall deviations, for scenario 1.3's distributed aircrafts. CR stands for Conflict Resolution.

3.5.4 Case Scenario 1.4

$z = z_1$

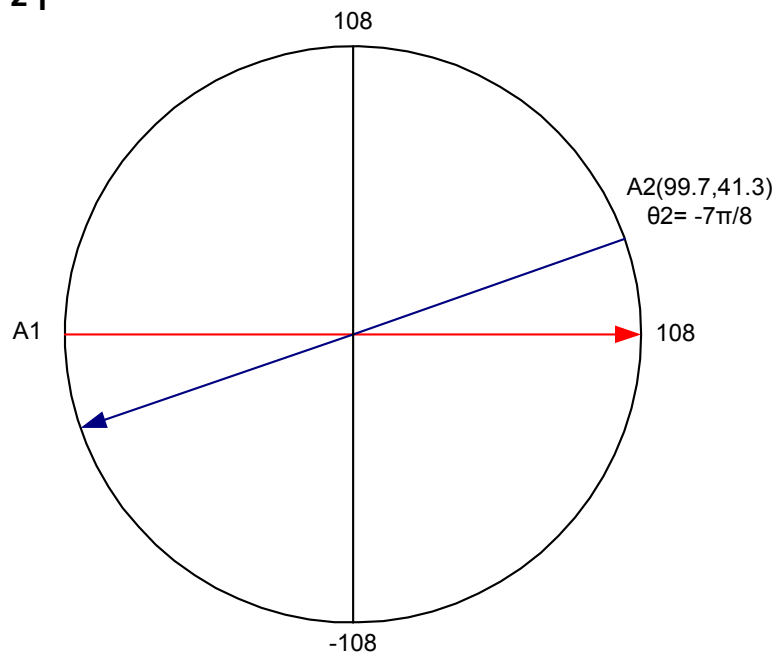


Figure 3.16. Scenario 1.4's distributed aircrafts and their trajectories, before conflict resolution. ($z=z_1$)

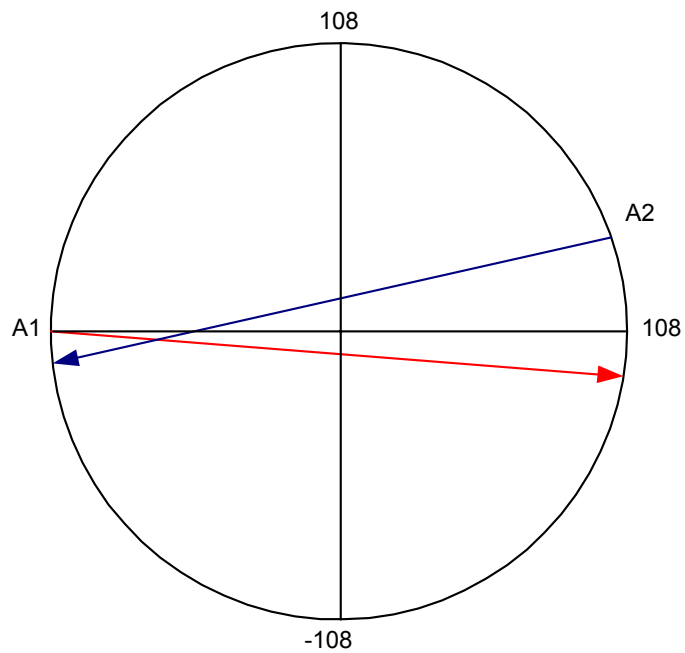


Figure 3.17. Scenario 1.4's distributed aircrafts and their trajectories, after conflict resolution. ($z=z_1$)

Scenario 1.4	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	-0.139	-0.374
Heading angle before CR (rad)	0	-2.747
Heading angle change after CR (rad)	-0.037	-0.048

Figure 3.18. Table showing velocities, heading angles and overall deviations, for scenario 1.4's distributed aircrafts. CR stands for Conflict Resolution.

3.5.5 Case Scenario 1.5

$$z = z_1$$

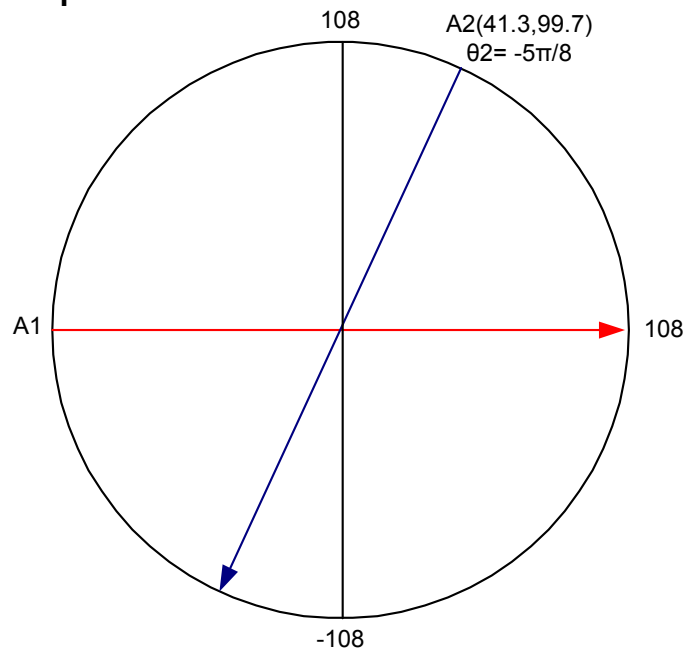


Figure 3.19. Scenario 1.5's distributed aircrafts and their trajectories, before conflict resolution. ($z=z_1$)

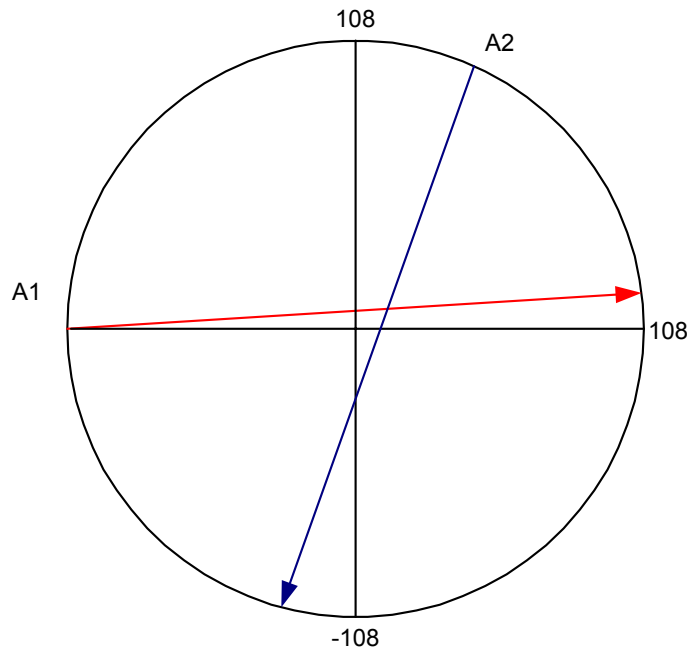


Figure 3.20. Scenario 1.5's distributed aircrafts and their trajectories, after conflict resolution. ($z=z_1$)

Scenario 1.5	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	0.198	0.535
Heading angle before CR (rad)	0	-1.962
Heading angle change after CR (rad)	0.041	0.058

Figure 3.21. Table showing velocities, heading angles and overall deviations, for scenario 1.5's distributed aircrafts. CR stands for Conflict Resolution.

3.5.6 Case Scenario 1.6

$z = z_1$

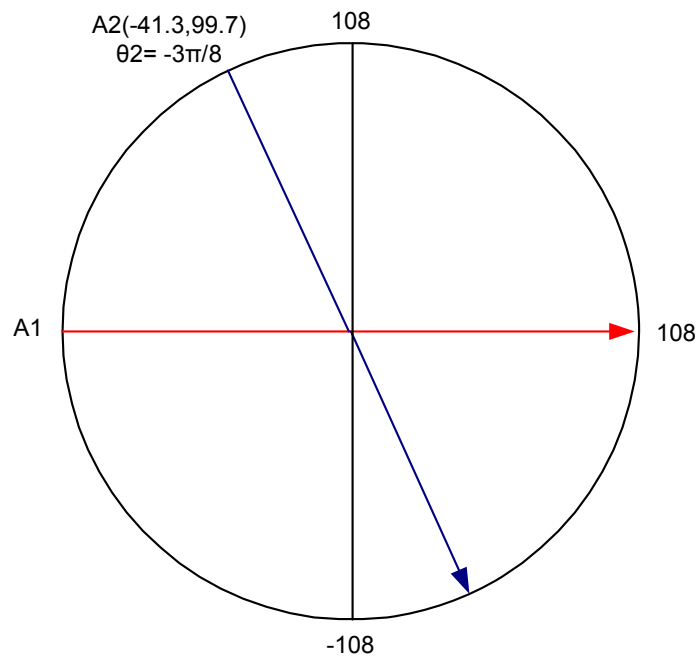


Figure 3.22. Scenario 1.6's distributed aircraft and their trajectories, before conflict resolution. ($z=z_1$)

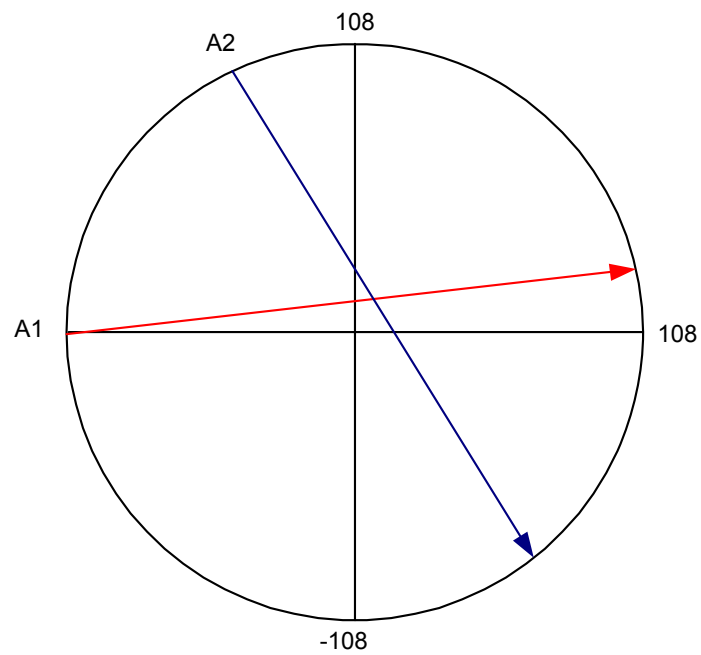


Figure 3.23. Scenario 1.6's distributed aircraft and their trajectories, after conflict resolution. ($z=z_1$)

Scenario 1.6	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	-0.773	0.298
Heading angle before CR (rad)	0	-1.177
Heading angle change after CR (rad)	0.076	0.073

Figure 3.24. Table showing velocities, heading angles and overall deviations, for scenario 1.6's distributed aircrafts. CR stands for Conflict Resolution.

3.5.7 Case Scenario 1.7

$$z = z_1$$

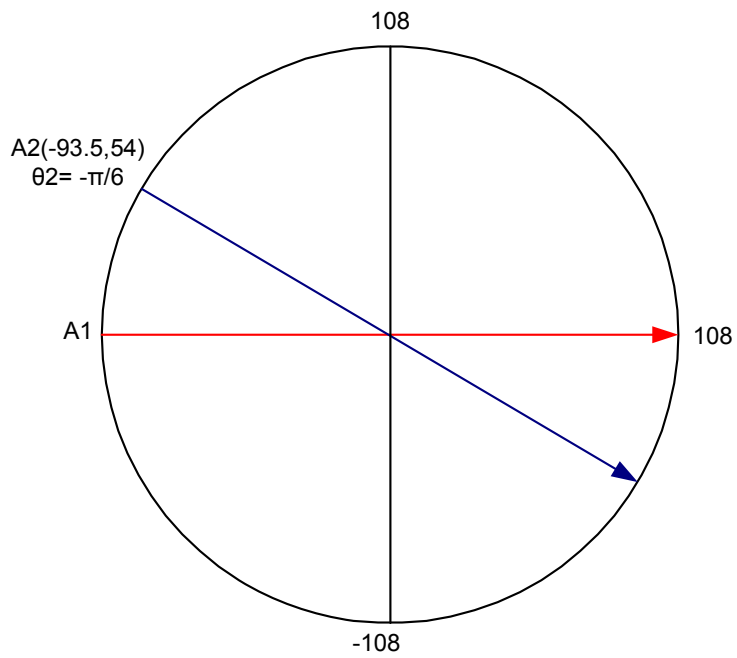


Figure 3.25. Scenario 1.7's distributed aircrafts and their trajectories, before conflict resolution. ($z=z_1$)

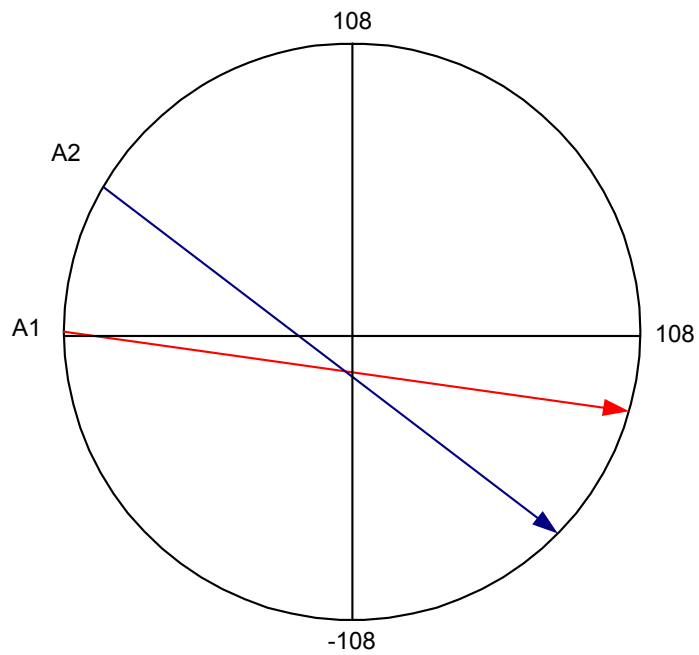


Figure 3.26. Scenario 1.7's distributed aircrafts and their trajectories, after conflict resolution. ($z=z_1$)

Scenario 1.7	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	0	-0.379
Heading angle before CR (rad)	0	-0.523
Heading angle change after CR (rad)	-0.170	-0.154

Figure 3.27. Table showing velocities, heading angles and overall deviations, for scenario 1.7's distributed aircrafts. CR stands for Conflict Resolution.

3.5.8 Case Scenario 1.8

$z = z_1$

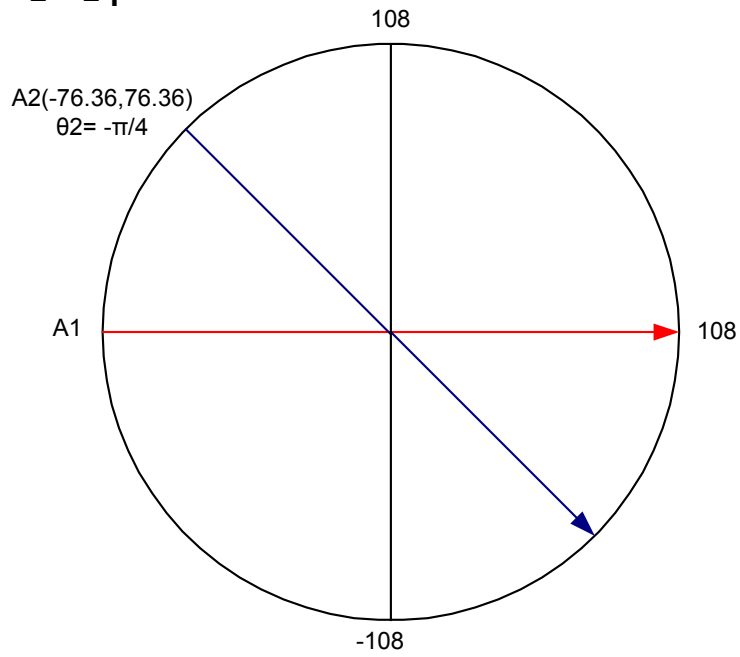


Figure 3.28. Scenario 1.8's distributed aircrafts and their trajectories, before conflict resolution. ($z=z_1$)

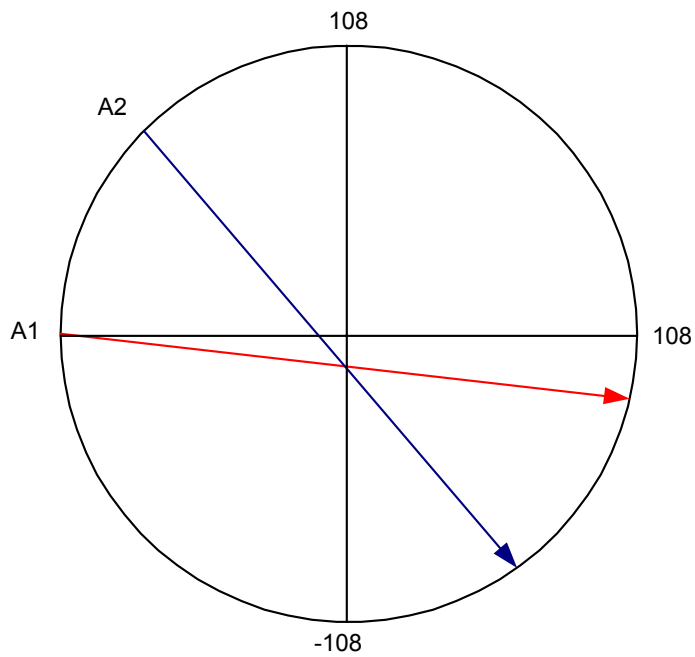


Figure 3.29. Scenario 1.8's distributed aircrafts and their trajectories, after conflict resolution. ($z=z_1$)

Scenario 1.8	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	0.318	-0.113
Heading angle before CR (rad)	0	-0.785
Heading angle change after CR (rad)	-0.116	-0.103

Figure 3.30. Table showing velocities, heading angles and overall deviations, for scenario 1.8's distributed aircrafts. CR stands for Conflict Resolution.

3.5.9 Case Scenario 2.1

$$z = z_2$$

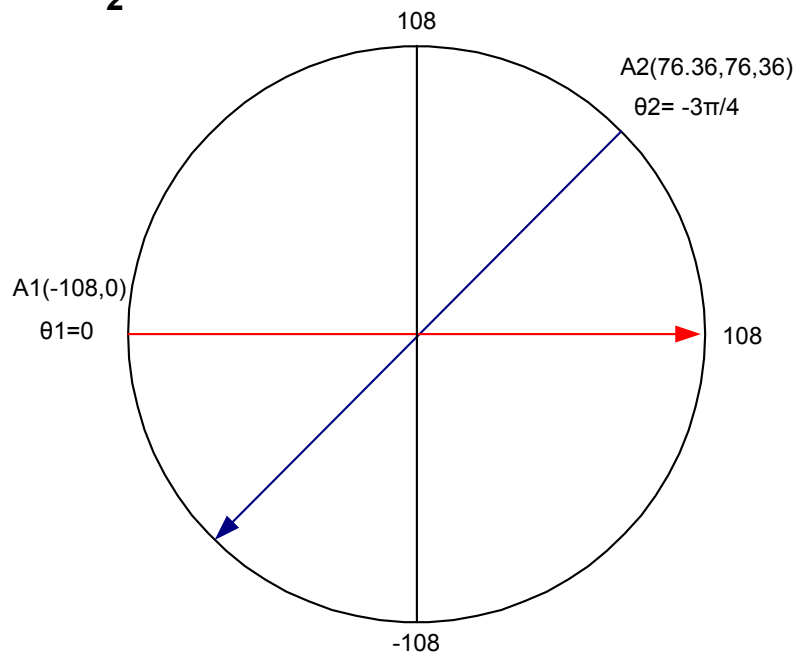


Figure 3.31.Scenario 2.1's distributed aircrafts and their trajectories, before conflict resolution.($z = z_2$)

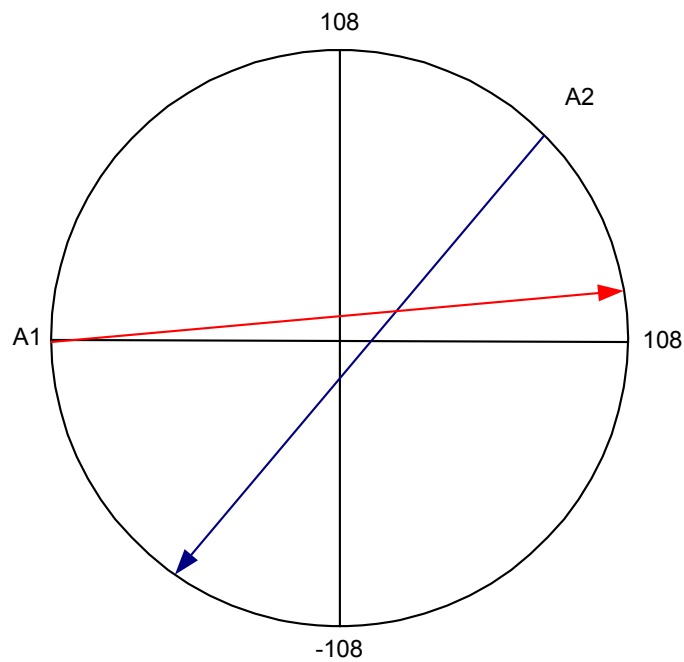


Figure 3.32.Scenario 2.1's distributed aircrafts and their trajectories, after conflict resolution.($z = z_2$)

Scenario 2.1	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	-0.1	0.1
Heading angle before CR(rad)	0	-2.355
Heading angle change after CR (rad)	0.045	0.045

Figure 3.33. Table showing the velocities, heading angles and overall deviations, for scenario 2.1's distributed aircrafts. CR stands for Conflict Resolution.

3.5.10 Case Scenario 2.2

$$z = z_2$$

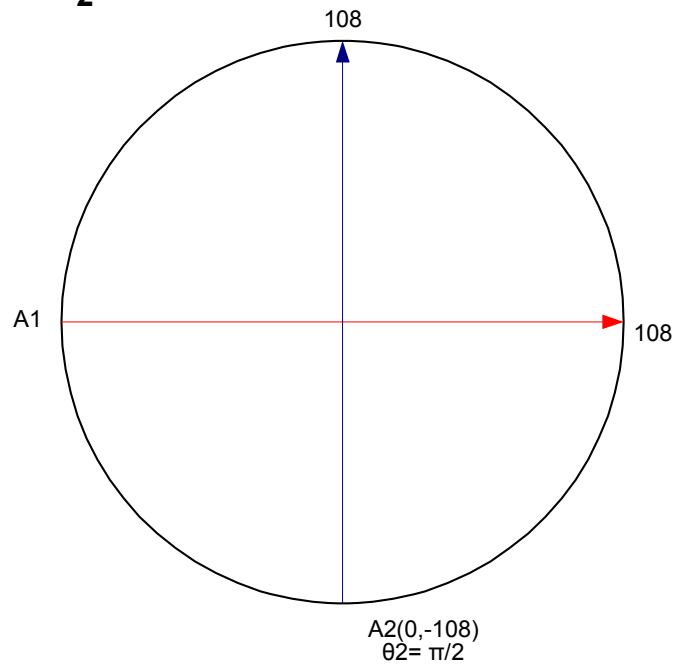


Figure 3.34. Scenario 2.2's distributed aircrafts and their trajectories, before conflict resolution. ($z = z_2$)

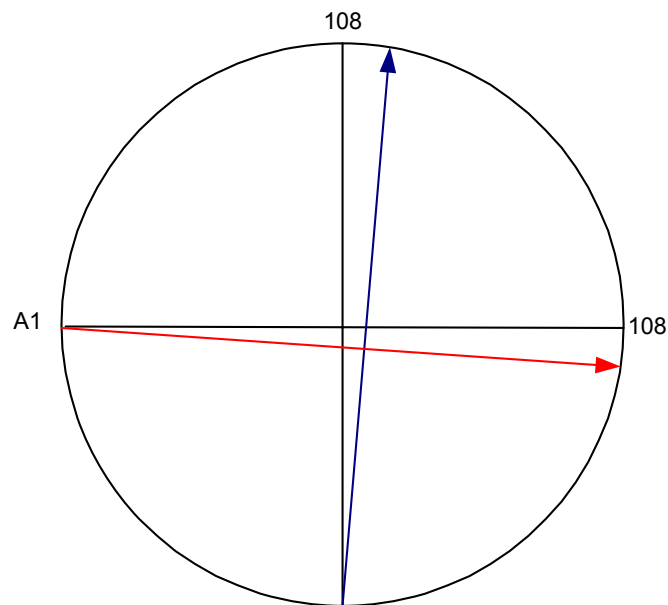


Figure 3.35. Scenario 2.2's distributed aircrafts and their trajectories, after conflict resolution. ($z = z_2$)

Scenario 2.2	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	-0.4	0.4
Heading angle before CR (rad)	0	1.57
Heading angle change after CR (rad)	-0.059	-0.059

Figure 3.36. Table showing velocities, heading angles and overall deviations, for scenario 2.2's distributed aircrafts. CR stands for Conflict Resolution.

3.5.11 Case Scenario 2.3
head to head
 $z = z_2$

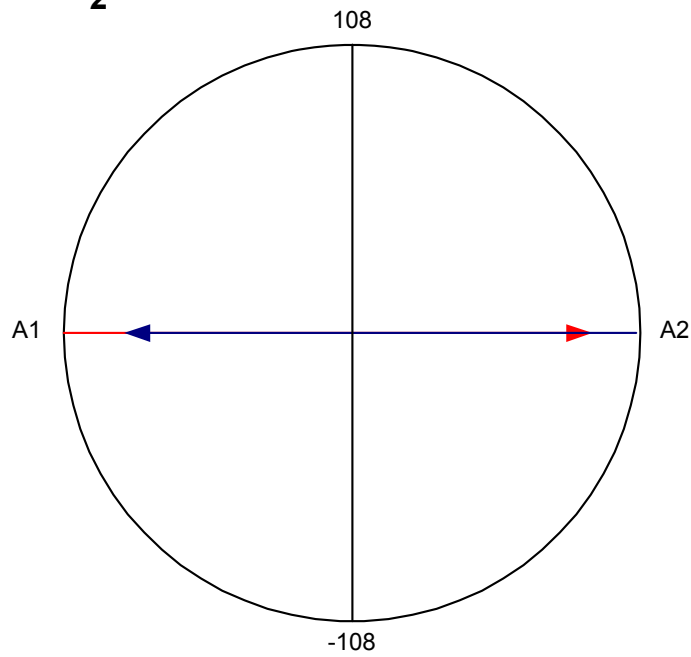


Figure 3.37.Scenario 2.3's distributed aircraft and their trajectories, before conflict resolution.($z = z_2$)

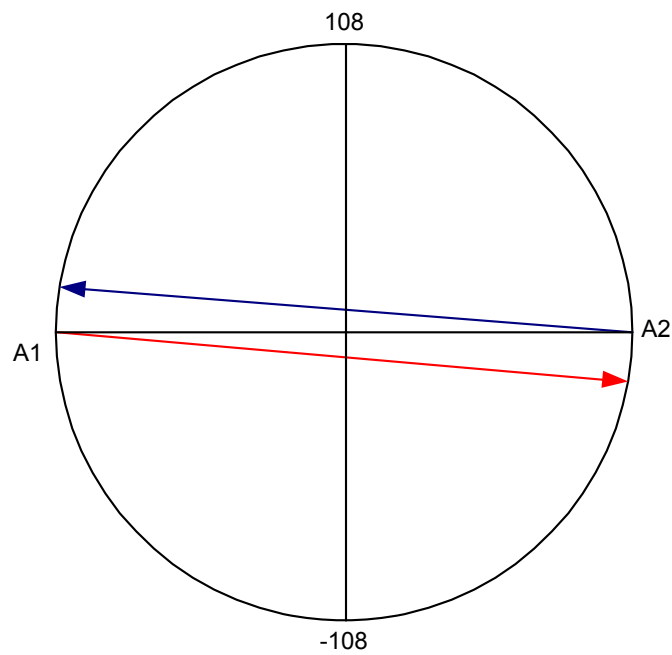


Figure 3.38.Scenario 2.3's distributed aircraft and their trajectories, after conflict resolution.($z = z_2$)

Scenario 2.3	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	-0.241	-0.488
Heading angle before CR (rad)	0	3.14
Heading angle change after CR (rad)	-0.042	-0.042

Figure 3.39. Table showing velocities, heading angles and overall deviations, for scenario 2.3's distributed aircrafts. CR stands for Conflict Resolution.

3.5.12 Case Scenario 2.4

$$z = z_2$$

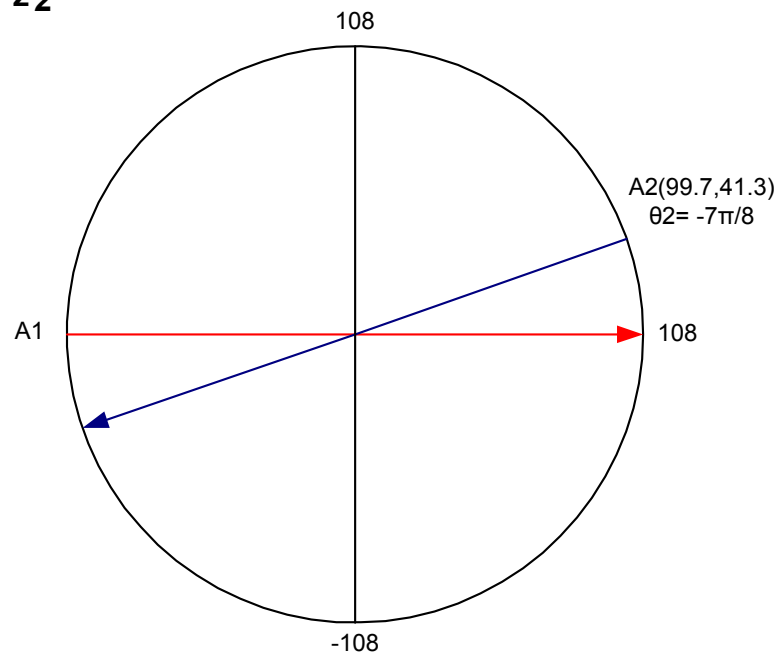


Figure 3.40. Scenario 2.4's distributed aircrafts and their trajectories, before conflict resolution. ($z = z_2$)

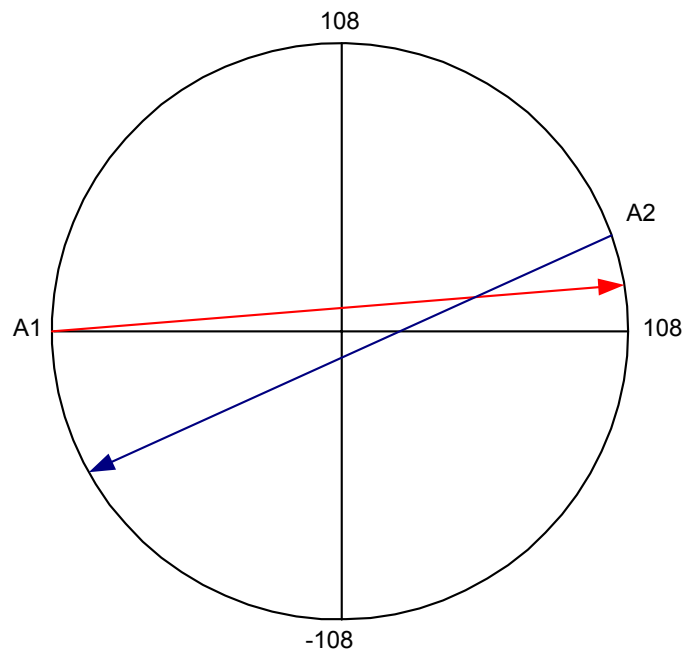


Figure 3.41. Scenario 2.4's distributed aircrafts and their trajectories, after conflict resolution. ($z = z_2$)

Scenario 2.4	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	-0.267	0.268
Heading angle before CR (rad)	0	-2.747
Heading angle change after CR (rad)	0.042	0.042

Figure 3.42. Table showing velocities, heading angles and overall deviations, for scenario 2.4's distributed aircrafts. CR stands for Conflict Resolution.

3.5.13 Case Scenario 2.5

$$z = z_2$$

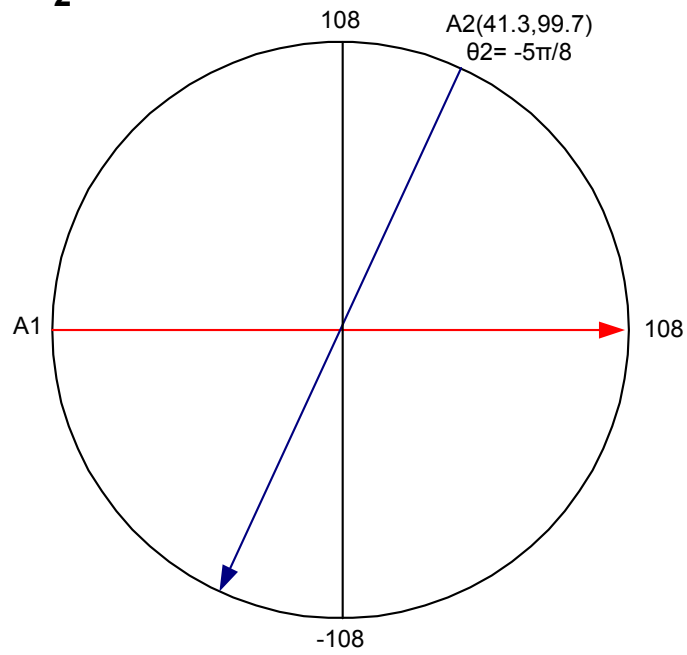


Figure 3.43. Scenario 2.5's distributed aircrafts and their trajectories, before conflict resolution. ($z = z_2$)

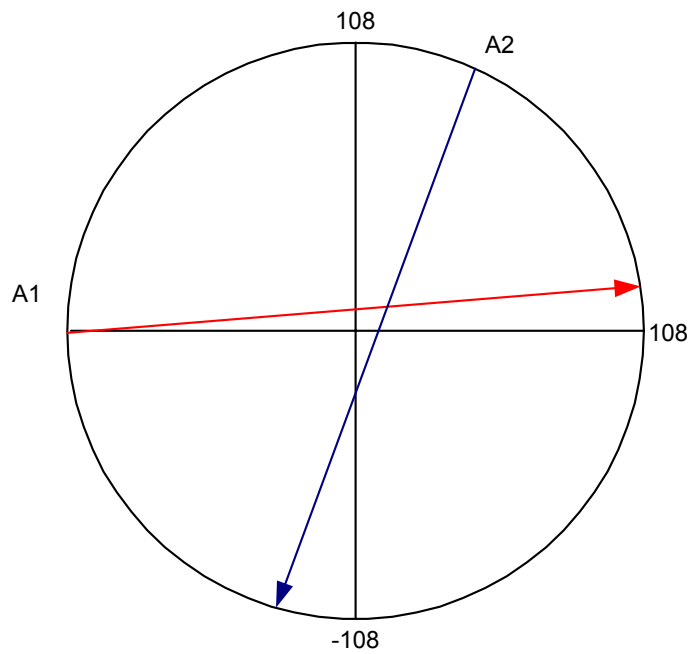


Figure 3.44. Scenario 2.5's distributed aircrafts and their trajectories, after conflict resolution. ($z = z_2$)

Scenario 2.5	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	-0.2	0.2
Heading angle before CR (rad)	0	-1.962
Heading angle change after CR (rad)	0.050	0.050

Figure 3.45. Table showing velocities, heading angles and overall deviations, for scenario 2.5's distributed aircrafts. CR stands for Conflict Resolution.

3.5.14 Case Scenario 2.6

$z = z_2$

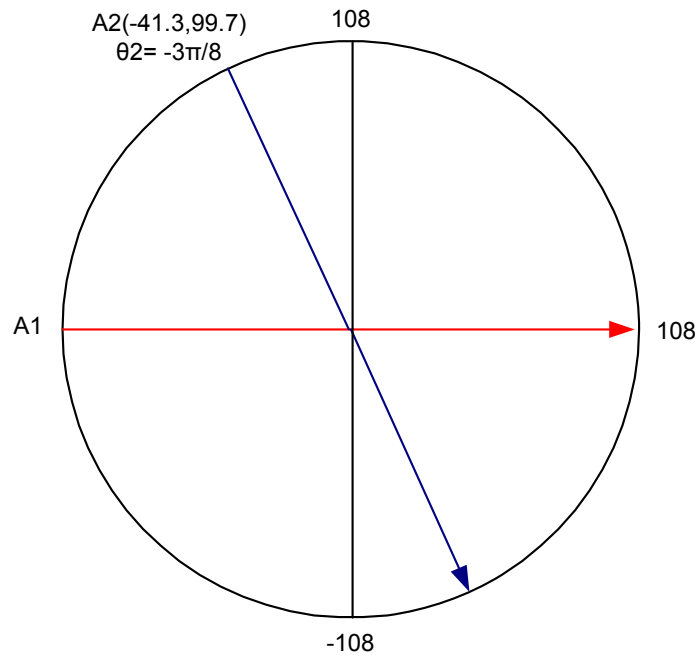


Figure 3.46. Scenario 2.6's distributed aircraft and their trajectories, before conflict resolution. ($z = z_2$)

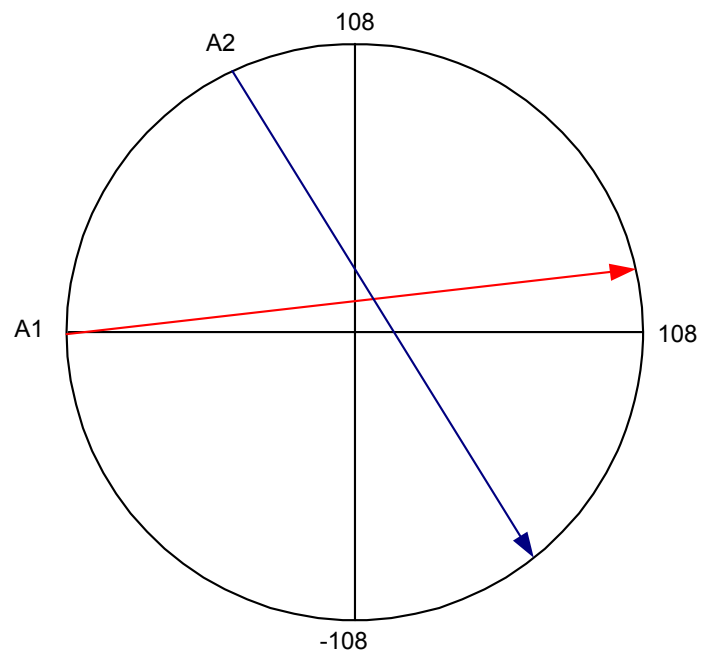


Figure 3.47. Scenario 2.6's distributed aircraft and their trajectories, after conflict resolution. ($z = z_2$)

Scenario 2.6	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	-0.35	0.35
Heading angle before CR (rad)	0	-1.177
Heading angle change after CR (rad)	0.074	0.074

Figure 3.48. Table showing velocities, heading angles and overall deviations, for scenario 2.6's distributed aircrafts. CR stands for Conflict Resolution.

3.5.15 Case Scenario 2.7

$$z = z_2$$

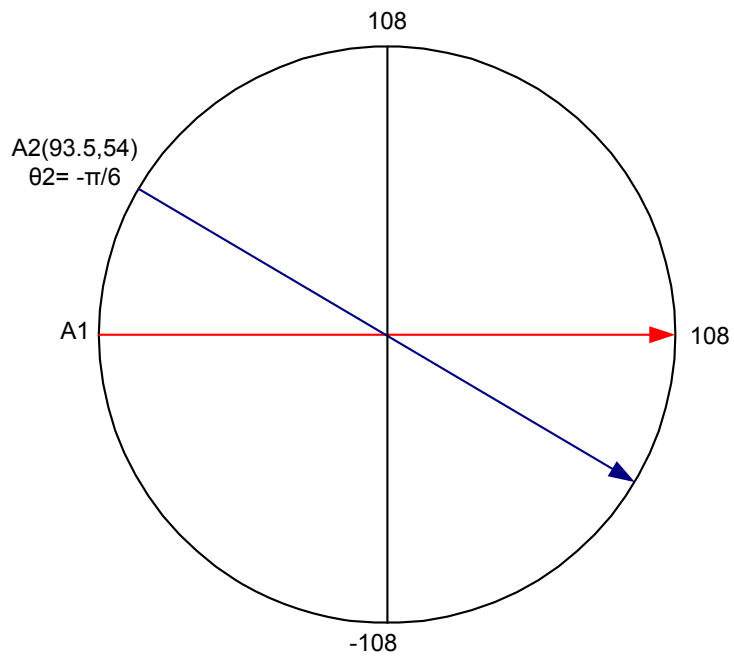


Figure 3.49. Scenario 2.7's distributed aircrafts and their trajectories, before conflict resolution. ($z = z_2$)

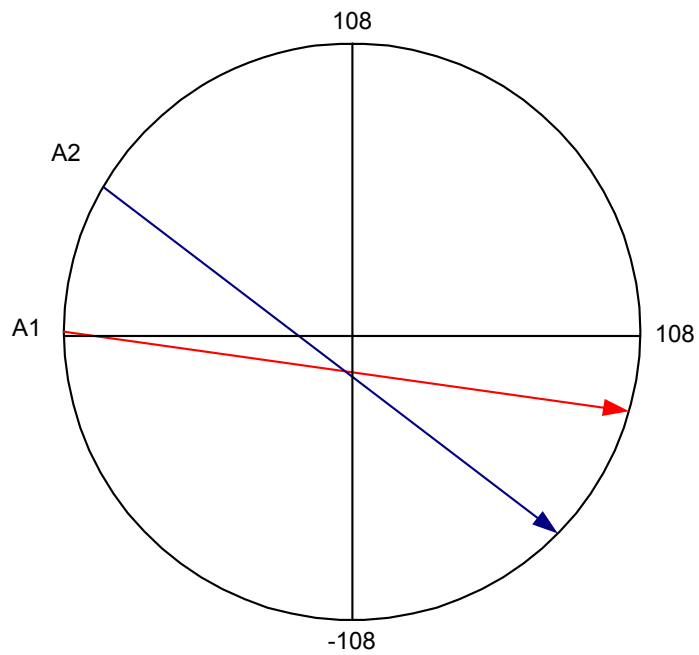


Figure 3.50. Scenario 2.7's distributed aircrafts and their trajectories, after conflict resolution. ($z = z_2$)

Scenario 2.7	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	0.390	-0.390
Heading angle before CR (rad)	0	-0.523
Heading angle change after CR (rad)	-0.158	-0.146

Figure 3.51. Table showing velocities, heading angles and overall deviations, for scenario 2.7's distributed aircrafts. CR stands for Conflict Resolution.

3.5.15 Case Scenario 2.8

$$z = z_2$$

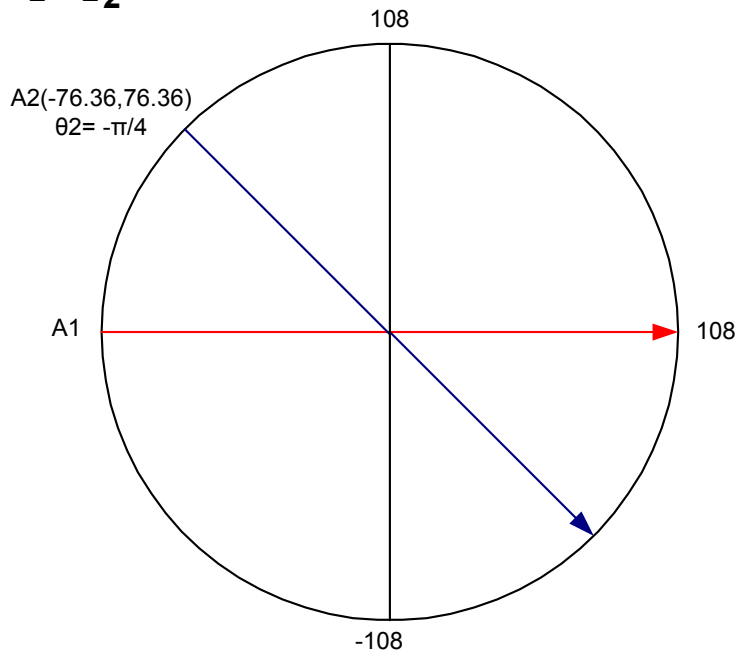


Figure 3.52.Scenario 2.8's distributed aircrafts and their trajectories, before conflict resolution.($z = z_2$)

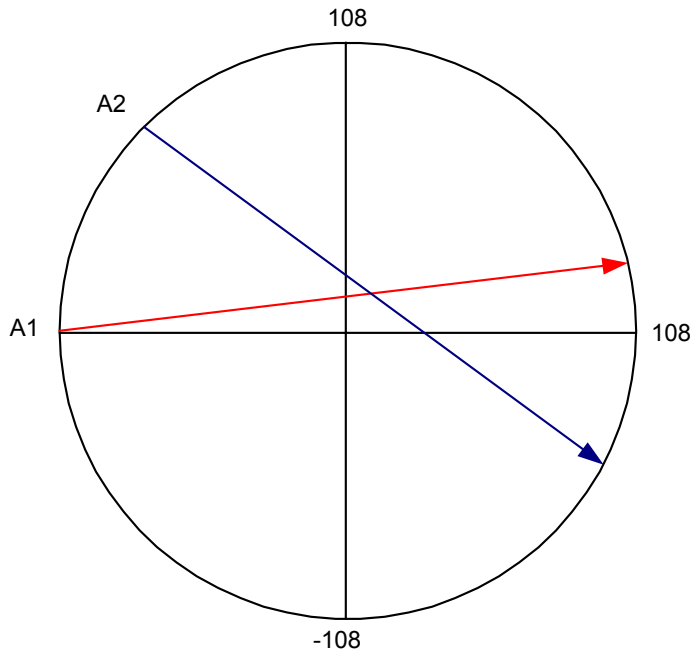


Figure 3.53.Scenario 2.8's distributed aircrafts and their trajectories, after conflict resolution.($z = z_2$)

Scenario 2.8	Aircraft 1	Aircraft 2
Velocity before CR (km/min)	15	15
Velocity change after CR (km/min)	-0.170	0.170
Heading angle before CR (rad)	0	-0.785
Heading angle change after CR (rad)	0.105	0.107

Figure 3.54. Table showing velocities, heading angles and overall deviations, for scenario 2.8's distributed aircrafts. CR stands for Conflict Resolution.

3.7 Conclusive Results

At first, it is necessary to discuss the issue of the computational time, the time it took for GAMS to obtain the previously presented results. We can safely say that for all the previous plots we presented, executional time was between the space of 2-4 seconds, per simulation. We find this time period satisfactory, because of the non-linearities introduced to the problem and makes us optimistic of potential use of this software in real time situations.

Secondly, we will present a table showing the sum of the absolute values of the velocity and heading angle deviations seperately, for all case scenarios, for both objective functions we chose to minimize (z1,z2).

Sc.	$\Delta u(z1)$	$\Delta \theta(z1)$	$\Delta u(z2)$	$\Delta \theta(z2)$
1	0.31	0.091	0.20	0.090
2	1.57	0.118	0.80	0.118
3	0.354	0.083	0.729	0.084
4	0.513	0.085	0.520	0.084
5	0.733	0.099	0.800	0.1
6	1.07	0.149	0.70	0.148
7	0.379	0.324	0.78	0.304
8	0.431	0.219	0.34	0.212

Figure 3.55. Table showing the absolut sum of the velocity and heading angle deviations for z1 and z2 simulations.

In the above figure we can notice that the differences in the obtained values between the two different objective functions are quite small. In fact they seem to agree with each other as the resolution of the conflict is concerned, at least in pure numbers, although there are differences for example in the heading angle deviations, z_1 decides left turns in one case scenario while z_2 decides right turns. It is quite interesting, the choice of the cost function, because it gives a potential client of the software the ability to “suit” the problem to his own needs, for example, a client may want to minimize the total flight time, so he chooses an objective function that represents the aircrafts total flight time in order to minimize it.

Appendix A

Basic Features of GAMS

Some basic features of GAMS are explained in the following subsections.

A.1 General Principles

The design of GAMS has incorporated ideas drawn from relational database theory and mathematical programming and has attempted to merge these ideas to suit the needs of strategic modelers. Relational database theory provides a structured framework for developing general data organization and transformation capabilities. Mathematical programming provides a way of describing a problem and a variety of methods for solving it. The following principles were used in designing the system:

- All existing algorithmic methods should be available without changing the user's model representation. Introduction of new methods, or of new implementation of existing methods, should be possible without requiring changes in existing models. Linear, nonlinear, mixed integer, mixed integer nonlinear optimization problems can currently be accommodated.
- The optimization problem should be independent of the data it uses. This separation of logic and data allows a problem to be increased in size without causing an increase in the complexity of the representation.
- The use of relational requires that the allocation of computer resources be automated. This means that large and complex models can be constructed without the user having to worry details such as array sizes and scratch storage.

A.2 Documentation

The GAMS model representation is in a form that can be easily read by people and by computers. This means that the GAMS program itself, is the documentation of the model, and that the separate description required in the past (which was a burden to maintain) is no longer needed. Moreover, the design of GAMS incorporates the following features that specifically address the user's documentation needs:

- A GAMS model representation is concise, and makes full use of the elegance of mathematical representation.
- All data transformations are specified concisely and algebraically. This means that all data can be entered in their most elemental form and that all transformations made in constructing the model and in reporting are available for inspection.
- Explanatory text can be made part of the definition of all symbols and is reproduced whenever associated values are displayed.

Of course, some discipline is needed to take full advantage of these design features, but the aim is to make models more accessible, more understandable, more verifiable and hence more credible.

A.3 Portability

The GAMS system is designed so that models can be solved on different types of computers with no change. A model developed on a small personal computer can later be solved on a large mainframe. One person can develop a model that is later used by others who maybe physically distant from the original developer. In contrast to previous approaches, only one document need to be moved—the GAMS statement of the model. It contains all the data and logical specifications needed to solve the model.

A.4 User Interface

Portability concerns also have implications on the user interface. The basic GAMS system, is file-oriented, and no special editor or graphical input and output routines exist. Rather than burden, the user with having to learn yet another set of editing commands, GAMS offers an open architecture in which each user can use his word processor or editor of choice. This basic user interface facilitates the integration of GAMS with a variety of existing user environments.

A.5 Model Library

From the early stages of developing GAMS, models have been collected to be used in a library of examples. Many of these are standard textbook examples and can be used in problem formulation or to illustrate points about GAMS. Others are models that can be used in policy or sector analysis and are interesting for both the data and the method they use. A collection of models is now included in all GAMS systems, along with a database to help users locate examples that covers all his basic interests.

Appendix B

Mixed Integer NonLinear Programming

Recently, the area of Mixed Integer Nonlinear Programming (MINLP) has experienced tremendous growth and a flourish of research activity. We will give a brief overview of past developments in the MINLP arena and discuss some of the future work that can aid the development of MINLP.

B.1 Introduction

Mixed Integer Nonlinear Programming (MINLP) refers to mathematical programming with continuous and discrete variables and nonlinearities in the objective function and constraints. The use of MINLP is a natural approach of formulating problems where it is necessary to simultaneously optimize the system structure (discrete) and parameters (continuous).

MINLPs have been used in various applications, including the process industry and the financial, engineering, management science and operations research sectors. It includes problems in process flow sheets, portfolio selection, batch processing in chemical engineering (consisting of mixing, reaction, and centrifuge separation), and optimal design of gas or water transmission networks. Other areas of interest include the automobile, aircraft, and VLSI manufacturing areas. The needs in such diverse areas have motivated research and development in MINLP solver technology, particularly in algorithms for handling large-scale, highly combinatorial and highly nonlinear problems.

The general form of a MINLP is:

$$\begin{array}{ll} \text{minimize} & f(x, y) \\ \text{subject to} & g(x, y) \leq 0 \\ & x \in X \\ & y \in Y \quad \text{integer} \end{array} \tag{1}$$

The function $f(x, y)$ is a nonlinear objective function and $g(x, y)$ a nonlinear constraint function. The variables x, y are the decision variables, where y is required to be integer valued. X and Y are bounding-box-type restrictions on the variables. We refer to [12] for more information about MINLP fundamentals in textbook format.

B.2 Algorithms

MINLP problems are precisely so difficult to solve, because they combine all the difficulties of both of their subclasses: the combinatorial nature of mixed integer programs (MIP) and the difficulty in solving nonconvex (and even convex) nonlinear programs (NLP). Because subclasses MIP and NLP are among the class of theoretically difficult problems (NP-complete), so it is not surprising that solving MINLP can be a challenging and daring venture. Fortunately, the component structure of MIP and NLP within MINLP provides a collection of natural algorithmic approaches, exploiting the structure of each of the subcomponents.

B.2.1 Solution Approaches

Methods for solving MINLPs include innovative approaches and related techniques taken and extended from MIP. Outer Approximation (OA) methods, Branch-and-Bound (B&B), Extended Cutting Plane methods, and Generalized Bender's Decomposition (GBD) for solving MINLPs have been discussed in the literature since the early 1980's. These approaches generally rely on the successive solutions of closely related NLP problems. For example, B&B starts out forming a pure continuous NLP problem by dropping the integrality requirements of the discrete variables (often called the relaxed MINLP or RMINLP). Moreover, each node of the emerging B&B tree represents a solution of the RMINLP with adjusted bounds on the discrete variables.

In addition, OA and GBD require the successive solution of a related MIP problem. Both algorithms decompose the MINLP into an NLP subproblem that has the discrete variables fixed and a linear MIP master problem. The main difference between GBD and OA is in the definition of the MIP master problem. OA relies on tangential planes (or linearizations), effectively reducing each subproblem to a smaller feasible set, whereas the master MIP problem generated by GBD is given by a dual representation of the continuous space.

The approaches described above only guarantee global optimality under (generalized) convexity. Deterministic algorithms for global optimization of nonconvex problems require the solution of subproblems obtained via convex relaxations of the original problem in a branch-and-bound context, and have been quite successful in solving MINLPs.

B.3 Software

Although theoretical algorithmic ideas for solving MINLP have been around for a while, the practical implementation of such concepts is much more difficult. Memory limitations, efficient numerical linear algebra routines, suitable algorithmic tolerances, and determining default solver options are some of the key issues faced when extending algorithms to large-scale, general-purpose software. In this section we give a brief and possibly incomplete historical overview of practical general purpose MINLP software.

B.3.1 Commercial MINLP Software Packages

The earliest commercial software package that could solve MINLP problems was SCICONIC in the mid 1970's. Rather than handling nonlinearities directly, linked SOS variables provided a mechanism to represent discretized nonlinear functions and allowed solving the problem via MIP. In the mid 1980's Grossman and Kocis developed GAMS/DICOPT, a general purpose MINLP algorithm based on the outer approximation method. In the early 1990's LINDOs and What's Best B&B code using the Generalized Reduced Gradient (GRG) code for subproblems was extended to solve MINLPs.

Since then a number of excellent academic as well as commercial codes have surfaced, including alphaECP and mittlp, both of which are based on extended cutting plane methods, and MINLP BB and SBB, which use branch-and-bound to solve relaxed NLP subproblems. Even on the frontier of global MINLP, reliable and large-scale packages have materialized including alphaBB and BARON, which use convex relaxations in a branch and bound framework.

B.3.2 Modelling Languages

The emergence of algebraic modeling languages in the mid to late 1980's and early 1990's has greatly simplified the process of modeling, in particular the formulation of MINLP type problems. Also, from a MINLP solver perspective, a modeling system delivers reliable black-box-type function evaluations and first and second order derivative information. Finally, the common solver interface of a modeling system allows MINLP algorithms to deploy existing NLP and MIP solvers to solve subproblems in a seamless way. A collection of MINLP models can be found in libraries such as MacMINLP (AMPL models), (GAMS models) and as a superset MINLPLib (GAMS models). The latter is available as part of the MINLP World. MINLP World is a forum for discussion and dissemination of information about all aspects of MINLP.

B.4 Recent Developments

With the recent progress made in global optimization, the importance of modeling

systems has taken on a more significant role. In particular, most global solvers require more than black-box function evaluations. These solvers need structural information of algebraic expressions to build convex relaxations. AlphaBB and the modeling language MINOPT, as well as the recent release of GAMS/BARON have shown the feasibility of this concept.

Another important advancement is the implementation of open algorithms. AIMMS-OA is an outer approximation method similar to GAMS/DICOPT, but with the distinct feature that it allows user modification for fine-tuning the method for a particular problem. Such an open approach allows advanced users to adjust the algorithm to suit the problem at hand.

Recent research has also focused on combining of Random Search (RS), such as Tabu, Scatter Search, Simulated Annealing or Genetic Algorithms, with NLP methods. Recent implementations like OQNLP and LaGO have proven to be quite successful.

Finally, the area of Disjunctive Programming uses disjunctions and logic propositions to represent the discrete decisions in the continuous and discrete space respectively. Disjunctive programs, conveniently modeled and automatically reformulated in big M or convex region models, give access to a rich area of applications. Widespread interest in such models has spawned a new computing environment (LogMIP), developed specifically for generalized disjunctive programming.

B.5 Future Directions

Progress in the MINLP arena has been significant in recent years, and we are now able to solve large-scale problems efficiently using a wide variety of approaches. However, MINLP has yet to reach the level of maturity that MIP has achieved. While the MIP community has benefited greatly from preprocessing to reduce model sizes and to detect special structure, MINLP technology is still lagging behind. NLP and MINLP preprocessing, similar to global methods, will require the delivery of structural information from the modeling languages. Progress on reliable large-scale NLP codes with restarting capabilities will have an immediate impact on MINLP.

Furthermore, combining individual algorithms (e.g. branch-and-bound and extended cutting plane method) with sophisticated search strategies (e.g. non-trivial B&B selection strategies) and heuristics to quickly determine integer solutions will help to close the gap. If research and development continues at the current level of activity, MINLP will soon achieve a stage of maturity enjoyed by the other areas in mathematical programming.

Appendix C

Software Codes and relative Documentation

As noted in the previous sections, the software codes were produced in the GAMS environment, a very useful tool when dealing with mathematical programming problems. In our case, we will present the software codes of case scenario 1 and 3(head to head) for both z_1 and z_2 objective functions, because the difference in the codes of all case scenarios is their initial configuration points. We will comment only for the software code of case 3 for the same reason.

GAMS software code for case scenario 1.1($z=z_1$)

**Declaration of basic parameters of the problem
Parameters $\pi, \omega, x_1, x_2, \psi_1, \psi_2, \theta_1, \theta_2, d,$
 $\alpha, \text{temp}, l, r, u_1, u_2, M ;$*

*$\pi=3.14159 ;$
*Minimum safe distance in Km
 $d=9;$
*Large number
 $M=50;$
*Initial velocities in Km/min
 $u_1=15;$
 $u_2=15;$
*Aircraft 1 configuration points
 $x_1=-108;$
 $\psi_1=0;$
 $\theta_1=0;$
*Aircraft 2 configuration points
 $x_2=76.36;$
 $\psi_2=76.36;$
 $\theta_2=-3*(\pi/4);$*

Definition of variables $\omega, \alpha, l, r, \text{temp}$, used in constrains

$$\begin{aligned}\omega &= \arctan((\psi_1 - \psi_2)/(x_1 - x_2)); \\ \alpha &= \sqrt{(x_1 - x_2)^2 + (\psi_1 - \psi_2)^2} / (\psi_1 - \psi_2); \\ \text{temp} &= d/\alpha; \\ l &= \omega + \arctan(\text{temp}/(\sqrt{1 - \text{temp}^2})); \\ r &= \omega - \arctan(\text{temp}/(\sqrt{1 - \text{temp}^2}));\end{aligned}$$

Variable z, q_1, q_2, p_1, p_2 ;

Binary variable y_1, y_2, y_3 ;

* y_1 controls which of A6 and A7 holds

* y_2 controls which of A9 and A10 holds

* y_3 controls which of the sets [A5,A6,A7] and [A8,A9,A10] holds

Equations

deviation, A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14;

$$q_{1,l} = 0.001;$$

$$q_{2,l} = 0.001;$$

$$p_{1,l} = 0.001;$$

$$p_{2,l} = 0.001;$$

*Objective function

$$\text{deviation.. } z = e = \text{abs}(q_1) + \text{abs}(q_2) + \text{abs}(p_1) + \text{abs}(p_2);$$

$$A1.. \quad q_1 = l = 15.66 - u_1;$$

$$A2.. \quad q_2 = l = 15.66 - u_2;$$

$$A3.. \quad -q_1 = l = u_1 - 14.4;$$

$$A4.. \quad -q_2 = l = u_2 - 14.4;$$

$$A5.. \quad (u_1 + q_1) \cos(\theta_1 + p_1) - (u_2 + q_2) \cos(\theta_2 + p_2) - M y_3 = l = 0;$$

$$A6.. \quad (u_1 + q_1) \sin(\theta_1 + p_1) - (u_1 + q_1) \cos(\theta_1 + p_1) \sin(l) \\ / \cos(l) - (u_2 + q_2) \sin(\theta_2 + p_2) + (u_2 + q_2) \cos(\theta_2 + p_2) \sin(l) \\ / \cos(l) - M y_1 - M y_3 = l = 0;$$

$$A7.. \quad -(u_1 + q_1) \sin(\theta_1 + p_1) + (u_1 + q_1) \cos(\theta_1 + p_1) \sin(r) \\ / \cos(r) + (u_2 + q_2) \sin(\theta_2 + p_2) - (u_2 + q_2) \cos(\theta_2 + p_2) \sin(r) \\ / \cos(r) + M y_1 - M y_3 = l = M;$$

$$A8.. \quad (u_1 + q_1) \cos(\theta_1 + p_1) + (u_2 + q_2) \cos(\theta_2 + p_2) + M y_3 = l = M;$$

$$A9.. \quad -(u_1 + q_1) \sin(\theta_1 + p_1) + (u_1 + q_1) \cos(\theta_1 + p_1) \sin(l) \\ / \cos(l) + (u_2 + q_2) \sin(\theta_2 + p_2) - (u_2 + q_2) \cos(\theta_2 + p_2) \sin(l) \\ / \cos(l) - M y_2 + M y_3 = l = M;$$

$$A10.. \quad (u_1 + q_1) \sin(\theta_1 + p_1) - (u_1 + q_1) \cos(\theta_1 + p_1) \sin(r) \\ / \cos(r) - (u_2 + q_2) \sin(\theta_2 + p_2) + (u_2 + q_2) \cos(\theta_2 + p_2) \sin(r) \\ / \cos(r) + M y_2 + M y_3 = l = 2 * M;$$

```

A11..    p1=l=0.17;
A12..    p2=l=0.17;
A13..    -p1=l=0.17;
A14..    -p2=l=0.17;

```

```

Model vc /all/ ;
option rminlp=conopt2;
option nlp=conopt2;
option mip=cplex;
option minlp=dicopt;
solve vc minimizing z using rminlp;
solve vc minimizing z using minlp;

```

GAMS software code for case scenario 1.3(z=z1)

```

*Declaration of basic parameters of the problem
Parameters pi,omega,x1,x2,psi1,psi2,theta1,theta2,d,
           alpha,temp,l,r,u1,u2,M ;

pi=3.14159 ;
*Minimum safe distance in Km
d=9;
*Large number
M=50;
*Initial velocities in Km/min
u1=15;
u2=15;
*Aircraft 1 configuration points
x1=-108;
psi1=0;
theta1=0;
*Aircraft 2 configuration points
x2=108;
psi2=0;
theta2=pi;

*Definition of variables omega,alpha,l,r,temp, used in constrains
omega=arctan((psi1-psi2)/(x1-x2));

```


$\alpha = \sqrt{(x1-x2)^2 + (psi1-psi2)^2}$;
 $temp = d/\alpha$;
 $l = \omega + \arctan(temp/\sqrt{1-temp*temp})$;
 $r = \omega - \arctan(temp/\sqrt{1-temp*temp})$;

Variable $z, q1, q2, p1, p2$;

Binary variable $y1, y2, y3$;

* $y1$ controls which of $A6$ and $A7$ holds

* $y2$ controls which of $A9$ and $A10$ holds

* $y3$ controls which of the sets $[A5, A6, A7]$ and $[A8, A9, A10]$ holds

Equations

deviation, $A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A14$;

$q1.l = 0.001$;

$q2.l = 0.001$;

$p1.l = 0.001$;

$p2.l = 0.001$;

*Objective function

deviation.. $z = e = \text{abs}(q1) + \text{abs}(q2) + \text{abs}(p1) + \text{abs}(p2)$;

$A1.. \quad q1 = l = 15.66 - u1$;

$A2.. \quad q2 = l = 15.66 - u2$;

$A3.. \quad -q1 = l = u1 - 14.4$;

$A4.. \quad -q2 = l = u2 - 14.4$;

$A5.. \quad (u1 + q1) * \cos(\theta1 + p1) - (u2 + q2) * \cos(\theta2 + p2) - M * y3 = l = 0$;

$A6.. \quad (u1 + q1) * \sin(\theta1 + p1) - (u1 + q1) * \cos(\theta1 + p1) * \sin(l) / \cos(l) - (u2 + q2) * \sin(\theta2 + p2) + (u2 + q2) * \cos(\theta2 + p2) * \sin(l) / \cos(l) - M * y1 - M * y3 = l = 0$;

$A7.. \quad -(u1 + q1) * \sin(\theta1 + p1) + (u1 + q1) * \cos(\theta1 + p1) * \sin(r) / \cos(r) + (u2 + q2) * \sin(\theta2 + p2) - (u2 + q2) * \cos(\theta2 + p2) * \sin(r) / \cos(r) + M * y1 - M * y3 = l = M$;

$A8.. \quad (u1 + q1) * \cos(\theta1 + p1) + (u2 + q2) * \cos(\theta2 + p2) + M * y3 = l = M$;

$A9.. \quad -(u1 + q1) * \sin(\theta1 + p1) + (u1 + q1) * \cos(\theta1 + p1) * \sin(l) / \cos(l) + (u2 + q2) * \sin(\theta2 + p2) - (u2 + q2) * \cos(\theta2 + p2) * \sin(l) / \cos(l) - M * y2 + M * y3 = l = M$;

$A10.. \quad (u1 + q1) * \sin(\theta1 + p1) - (u1 + q1) * \cos(\theta1 + p1) * \sin(r) / \cos(r) - (u2 + q2) * \sin(\theta2 + p2) + (u2 + q2) * \cos(\theta2 + p2) * \sin(r) / \cos(r) + M * y2 + M * y3 = l = 2 * M$;

```

A11..    p1=l=0.17;
A12..    p2=l=0.17;
A13..    -p1=l=0.17;
A14..    -p2=l=0.17;

```

```

Model vc /all/ ;
option rminlp=conopt2;
option nlp=conopt2;
option mip=cplex;
option minlp=dicopt;
solve vc minimizing z using rminlp;
solve vc minimizing z using minlp;

```

As we can clearly notice, the only difference between these two software codes is the configuration points of aircraft 2, since we didn't have to change aircraft 1's configuration points to obtain the case scenarios. With the code, helpful comments are displayed, nevertheless we are going to briefly analyze the software code.

At first, we declare the basic parameters used for solving the problem. We proceed in defining those parameters. We define the minimum safe distance between the two aircraft, the large number M used for the MINLP formulation, and the well known π . After defining also the initial configuration points of the aircrafts (initial velocities, heading angles, coordinates) we proceed in defining parameters used in VC and HAC methods for obtaining non-conflict constrains.

At this point, we formulate the necessary MINLP constrains, which GAMS refers to with the general term *equations*. To be more specific, equations A1-A4 and A11-A14 impose some bounds on velocities and heading angles respectively, while A5-A10 are the main constrains as we have formulate them in the corresponding section.

The program concludes with the commands that select the various solvers for the model. In our case, we firstly solve a relaxed version of the MINLP problem, in which the integer restrictions for variables y_1, y_2, y_3 do not apply. This allows the program to converge quickly around a small set of feasible solutions and then, after imposing the integer condition, to find more easily the desired values.

GAMS software code for case scenario 2.1(z=z2)

**Declaration of basic parameters of the problem*

*Parameters pi,omega,x1,x2,psi1,psi2,theta1,theta2,d,
alpha,temp,l,r,u1,u2,M ;*

pi=3.14159 ;

**Minimum safe distance in Km*

d=9;

**Large number*

M=50;

**Initial velocities in Km/min*

u1=15;

u2=15;

**Aircraft 1 configuration points*

x1=-108;

psi1=0;

theta1=0;

**Aircraft 2 configuration points*

x2=76.36;

psi2=76.36;

theta2=-3(pi/4);*

Definition of variables omega,alpha,l,r,temp, used in constrains

omega=arctan((psi1-psi2)/(x1-x2));

alpha=sqrt((x1-x2)(x1-x2)+(psi1-psi2)*(psi1-psi2)) ;*

temp=d/alpha;

*l=omega+arctan(temp/(sqrt(1-temp*temp)));*

*r=omega-arctan(temp/(sqrt(1-temp*temp)));*

Variable z,q1,q2,p1,p2;

Binary variable y1,y2,y3;

**y1 controls which of A6 and A7 holds*

**y2 controls which of A9 and A10 holds*

**y3 controls which of the sets [A5,A6,A7] and [A8,A9,A10] holds*

Equations

deviation,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,A13,A14;

q1.l=0.001;

q2.l=0.001;

p1.l=0.001;

p2.l=0.001;

**Objective function*

deviation.. $z=e=(p1*p1)+(p2*p2)+(q1*q1)+(q2*q2);$

A1.. $q1=l=15.66-u1;$

A2.. $q2=l=15.66-u2;$

A3.. $-q1=l=u1-14.4;$

A4.. $-q2=l=u2-14.4;$

A5.. $(u1+q1)*cos(theta1+p1)-(u2+q2)*cos(theta2+p2)-M*y3=l=0;$

A6.. $(u1+q1)*sin(theta1+p1)-(u1+q1)*cos(theta1+p1)*sin(l)$
 $/cos(l)-(u2+q2)*sin(theta2+p2)+(u2+q2)*cos(theta2+p2)*sin(l)$
 $/cos(l)-M*y1-M*y3=l=0;$

A7.. $-(u1+q1)*sin(theta1+p1)+(u1+q1)*cos(theta1+p1)*sin(r)$
 $/cos(r)+(u2+q2)*sin(theta2+p2)-(u2+q2)*cos(theta2+p2)*sin(r)$
 $/cos(r)+M*y1-M*y3=l=M;$

A8.. $(u1+q1)*cos(theta1+p1)+(u2+q2)*cos(theta2+p2)+M*y3=l=M;$

A9.. $-(u1+q1)*sin(theta1+p1)+(u1+q1)*cos(theta1+p1)*sin(l)$
 $/cos(l)+(u2+q2)*sin(theta2+p2)-(u2+q2)*cos(theta2+p2)*sin(l)$
 $/cos(l)-M*y2+M*y3=l=M;$

A10.. $(u1+q1)*sin(theta1+p1)-(u1+q1)*cos(theta1+p1)*sin(r)$
 $/cos(r)-(u2+q2)*sin(theta2+p2)+(u2+q2)*cos(theta2+p2)*sin(r)$
 $/cos(r)+M*y2+M*y3=l=2*M;$

A11.. $p1=l=0.17;$

A12.. $p2=l=0.17;$

A13.. $-p1=l=0.17;$

A14.. $-p2=l=0.17;$

Model vc /all/ ;

option rminlp=conopt2;

option nlp=conopt2;

option mip=cplex;

option minlp=dicopt;

solve vc minimizing z using rminlp;

solve vc minimizing z using minlp;

GAMS software code for case scenario 2.3(z=z2)

**Declaration of basic parameters of the problem*

*Parameters pi,omega,x1,x2,psi1,psi2,theta1,theta2,d,
alpha,temp,l,r,u1,u2,M ;*

pi=3.14159 ;

**Minimum safe distance in Km*

d=9;

**Large number*

M=50;

**Initial velocities in Km/min*

u1=15;

u2=15;

**Aircraft 1 configuration points*

x1=-108;

psi1=0;

theta1=0;

**Aircraft 2 configuration points*

x2=108;

psi2=0;

theta2=pi;

**Definition of variables omega,alpha,l,r,temp, used in constrains*

omega=arctan((psi1-psi2)/(x1-x2));

alpha=sqrt((x1-x2)(x1-x2)+(psi1-psi2)*(psi1-psi2)) ;*

temp=d/alpha;

*l=omega+arctan(temp/(sqrt(1-temp*temp)));*

*r=omega-arctan(temp/(sqrt(1-temp*temp)));*

Variable z,q1,q2,p1,p2;

Binary variable y1,y2,y3;

**y1 controls which of A6 and A7 holds*

**y2 controls which of A9 and A10 holds*

**y3 controls which of the sets [A5,A6,A7] and [A8,A9,A10] holds*

Equations

deviation,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,A13,A14;

q1.l=0.001;

q2.l=0.001;

p1.l=0.001;

p2.l=0.001;

**Objective function*

deviation.. $z=e=(p1*p1)+(p2*p2)+(q1*q1)+(q2*q2);$

A1.. $q1=l=15.66-u1;$

A2.. $q2=l=15.66-u2;$

A3.. $-q1=l=u1-14.4;$

A4.. $-q2=l=u2-14.4;$

A5.. $(u1+q1)*cos(theta1+p1)-(u2+q2)*cos(theta2+p2)-M*y3=l=0;$

A6.. $(u1+q1)*sin(theta1+p1)-(u1+q1)*cos(theta1+p1)*sin(l)$
 $/cos(l)-(u2+q2)*sin(theta2+p2)+(u2+q2)*cos(theta2+p2)*sin(l)$
 $/cos(l)-M*y1-M*y3=l=0;$

A7.. $-(u1+q1)*sin(theta1+p1)+(u1+q1)*cos(theta1+p1)*sin(r)$
 $/cos(r)+(u2+q2)*sin(theta2+p2)-(u2+q2)*cos(theta2+p2)*sin(r)$
 $/cos(r)+M*y1-M*y3=l=M;$

A8.. $(u1+q1)*cos(theta1+p1)+(u2+q2)*cos(theta2+p2)+M*y3=l=M;$

A9.. $-(u1+q1)*sin(theta1+p1)+(u1+q1)*cos(theta1+p1)*sin(l)$
 $/cos(l)+(u2+q2)*sin(theta2+p2)-(u2+q2)*cos(theta2+p2)*sin(l)$
 $/cos(l)-M*y2+M*y3=l=M;$

A10.. $(u1+q1)*sin(theta1+p1)-(u1+q1)*cos(theta1+p1)*sin(r)$
 $/cos(r)-(u2+q2)*sin(theta2+p2)+(u2+q2)*cos(theta2+p2)*sin(r)$
 $/cos(r)+M*y2+M*y3=l=2*M;$

A11.. $p1=l=0.17;$

A12.. $p2=l=0.17;$

A13.. $-p1=l=0.17;$

A14.. $-p2=l=0.17;$

Model vc /all/ ;

option rminlp=conopt2;

option nlp=conopt2;

option mip=cplex;

option minlp=dicopt;

solve vc minimizing z using rminlp;

solve vc minimizing z using minlp;

References

- [1] www.rtca.org/default.asp, *The Radio Technical Commission for Aeronautic Website*.
- [2] A. Bicchi and L. Pallottino. On Optimal Cooperative Conflict Resolution for Air Traffic Management Systems. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, December 2000.
- [3] L. Pallottino, E. Feron and A. Bicchi. Conflict Resolution Problems for Air Traffic Management Systems Solved with Mixed Integer Programming. *Journal*, 2001.
- [4] Z.-H. Mao, E. Feron and K. Bilimoria. Stability of Intersecting Aircraft n Flows Under Decentralized Conflict Avoidance Rules. *The American Institute of Aeronautics and Astronautics*, AIAA-2000-4271, 2001.
- [5] L. Pallottino, E. Feron and A. Bicchi. Mixed Integer Programming for Aircraft Conflict Resolution. *The American Institute of Aeronautics and Astronautics*, AIAA-2001-4295, 2001.
- [6] D. Dugail, Z.-H. Mao and E. Feron. Stability of Intersecting Aircraft Flows Under Centralized and Decentralized Conflict Avoidance Rules. *The American Institute of Aeronautics and Astronautics*, AIAA-2000-4296, 2001.
- [7] A. Richards, J. How, T. Schouwenaars and E. Feron. Plume Avoidance Maneuver Planning Using Mixed Integer Linear Programming. *The American Institute of Aeronautics and Astronautics*, AIAA-2000-4091, 2001.
- [8] W. P. Niedringhaus. Stream Option Manager (SOM): Automated Integration of Aircraft Separation, Merging, Stream Management and Other Air Traffic Control Functions. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 25, No. 9, September 1995.
- [9] C. Tomlin, G. J. Pappas and S. Sastry. Conflict Resolution for Air Tra±c Management: A Study in Multi-Agent Hybrid Systems. *IEEE Transactions on Automatic Control*, July 1998.

- [10] E. Frazzoli, Z.-H. Mao, J.-H. Oh and E. Feron. Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming. *Journal of Guidance, Control and Dynamics*, 2000.
- [11] www.gams.de, GAMS IDE 2.0.20.0. *GAMS User Guide*, 2002.
- [12] C. A. Floudas. *Nonlinear and Mixed-Integer Optimization*. Oxford University Press, 1995.
- [13] www.faa.gov, *The Federal Aviation Administration Website*.
- [14] *The Future of Air Traffic Control: Human Operators and Automation*. The National Academy Press, 1998.
- [15] J. M. Hoekstra. Designing for Safety: the Free Flight Air Traffic Management Concept, *Doctorate Thesis*, Delft University of Technology, Holland, 2001
- [16] www.aerosite.net
- [17] www.airtrafficcafe.com, *The Air Traffic Cafe Website*.

