

Algorithm for the aircraft trajectories considering No Fly Zones for a Flight Management System

Julien FAYS¹, Ruxandra Mihaela BOTEZ^{*:1}

^{*:1}Ecole de Technologie Supérieure, Laboratory of Applied Research in Active Controls, Avionics and AeroServoElasticity LARCASE
1100 Notre Dame West, Montreal, Que., Canada, H3C1K3
ruxandra.botez@etsmtl.ca*, jfays@estaca.fr

DOI: 10.13111/2066-8201.2013.5.3.9

Abstract: In this paper, optimal four dimensions trajectories (three space dimensions and one time dimension) are designed using new conceived algorithms, to connect all waypoints specified in the flight plan of an aircraft. The designs of these trajectories also include the management of No-Fly-Zones, which are zones over which flying is forbidden or temporarily refused. These zones can be either a meteorological obstacle to by-pass or zones in which air conflicts would be already too dense to allow the flying of a new aircraft.

Key Words: algorithm, aircraft trajectories, flight trajectory, avoidance method, optimization of the trajectory, meta-heuristic method

1. INTRODUCTION

In fact, two new algorithms are conceived and presented. The first algorithm is designed to by-pass the No-Fly-Zones while the second algorithm is used for the trajectory optimisation based on the meta-heuristic methodologies. This new second algorithm is combining the theories of two methods: *descent* and *taboo*, for the optimisation of trajectories. This algorithm was applied for the cruise flight between Paris and Montreal, for a Boeing 747-400 aircraft, where four NFZ were considered, and successfully by-passed. Two of these NFZ could be considered as problematic generators (the NFZ were considered to be on the aircraft trajectory, and it was important not to enter into these NFZ), while the two other NFZ were non-problematic generators (these NFZ were not on the trajectories, therefore there was no need in modifying the trajectory). A first non-problematic NFZ was considered below the flight trajectory, as if it was at the same altitude as the airplane, it was important to be avoided while the second NFZ was at the same altitude as the airplane, but not on the aircraft trajectory.

To validate that the trajectory between two successive samples as a line, the medium distance between two points will be given in percentage with respect to the total length of the airplane trajectory. The flight time will be compared with the real flight time values of Air France and Air Canada airlines. The real data in terms of fuel consumption and thrust will be compared with Boeing data.

A new code for the automatic following and optimization of a flight trajectory with autonomous avoidance of the NFZ is also here presented. This code proposes also the calculation of many flight data (such as the speed, the Estimated Arrival Time and the fuel consumption) that could be used for the real time flight following, the calculation of the Cost Index, or could serve as inputs to other aircraft embedded systems.

The Flight Management System will propose autonomously a new trajectory to the pilots that will take the decision to accept it or not. In case of the acceptance of the trajectory, this trajectory is sent to the Air Traffic Controller for its acceptance and if both parties agree, the information is further sent to the other airplanes. Therefore, the controllers concentrate further on the conflicts management. The main advantage of this work is the decrease of the Air-Traffic Controllers' workload.

2. PRESENTATION OF THE METHODOLOGIES

The NFZ (Non-Flying Zones) are inputs in the Flight Management System FMS, and are automatically dependent on the meteorological radar or radar of aerial traffic.

In the development period of simulation and validation, the NFZ are manually entered by the operator. The principle is to surround the NFZ by a circle with x_center_NFZ and y_centre_NFZ , that are the coordinates of the NFZ centre and r is its ray.

When the centre of the NFZ was seized, as well as the perimeter of the circle that defined it, the first version of the process of avoidance worked as follows. First of all, the algorithm verifies that no initial waypoint is present in the NFZ. In this case, a modification of the initial flight plan is required and the actions to take must be defined and authenticated by the professionals. Our computer program returns the information and requires a manual treatment manual of the situation. If this case is not presented, a NFZ is present between two points (for example, the points A and B) and the program creates automatically four points of avoidance around her. Their coordinates are the following:

- 1: $(x_centre_NFZ, y_centre_NFZ + 2r)$;
- 2: $(x_centre_NFZ, y_centre_NFZ - 2r)$;
- 3: $(x_centre_NFZ - 2r, y_centre_NFZ)$;
- 4: $(x_centre_NFZ + 2r, y_centre_NFZ)$.

The points are situated at a distance of $2r$ from the NFZ centre, in order to find themselves at a minimal distance r from NFZ, which allows its avoidance in all tested cases. It is a matter of a choice done during the development to test the program. Afterwards, it will be shown how this distance is reduced to optimize the avoidance system.

When the points of avoidance are arranged around the NFZ, the program automatically is testing all the possible flight trajectories between A and B with the aim to determine the most efficient one. These flight roads are composed from the point A , the point B and at least one of the four created avoidance points. The passage through at least one of these points allows the avoidance of the zone. In fact, the addition of one or several points of avoidance in the initial trajectory, therefore the change in the flight trajectory, allows the avoidance of the NFZ by the airplane.

Because of the fact that the rest of the flight plan is not modified, the airplane still remains on its initial trajectory. The four avoidance points are never used together, because it is more efficient to use a single point to bypass the NFZ.

As soon as one of these flight roads is validated, that is equivalent to say as soon as the proposed trajectory no longer cuts the NFZ the road is saved in the FMS memory.

Following the testing of all these roads, the validated roads are all measured and the shortest one is further preserved. The avoidance points are further added to the chain of waypoints. For example, if the preserved road uses the avoidance points 2 and 3, then the chain of waypoints to connect A to B (that is originally $[A, B]$ becomes $[A, 2, 3, B]$).

The method was tested and validated in the three possible situations:

- 1) A waypoint is in the NFZ. In this case, a manual treatment is required.

2) The trajectory crosses the NFZ. In this case, the automatic avoidance of the zone takes place.

3) The trajectory does not cross the NFZ. In this case, no corrective action is taken.

The following step is the optimization of the trajectory. It is necessary in fact to find a means to bypass the zone the most efficiently possible, while traversing a minimal distance, by using the same avoidance method, but with a different placement of the avoidance points.

The length d is defined as $d = r + n * step$ where n is the number of iterations needed to avoid the NFZ and $step$ is a fixed constant during the program development. In this development case the $step$ worth 0.1 length units. At the first iteration ($n = 1$), $d = r + 0.1$. At the second iteration ($n = 2$), $d = r + 0.2$, and to the tenth iteration ($n = 10$), $d = r + 1$.

The avoidance points are placed as follows: $(x_center_NFZ, y_centre_NFZ+d)$; $(x_centre_NFZ, y_centre_NFZ-d)$; $(x_centre_NFZ - d, y_center_NFZ)$; $(x_centre_NFZ + d, y_centre_NFZ)$. All the avoidance roads are tested at each iteration. If one of these roads is validated, selects the shortest validated road is selected at a certain iteration. Otherwise, n increases by a unity and the program tests again the possible different roads.

Thus, the found avoidance way passes very close to the NFZ by reducing the remaining distance to traverse. On figure 1, the ray of the NFZ (the big hatched circle) is of 3 length units and the avoidance waypoint is at 5.3 length units of the NFZ centre. The points of avoidance are therefore placed closer to the NFZ than in the non-optimised case, that shortens the avoidance trajectories. On this figure, one sees that the direct road, represented by stippled points, penetrates the NFZ. The algorithm of avoidance releases itself. The trajectory in full traits is the trajectory of avoidance selected by algorithm.

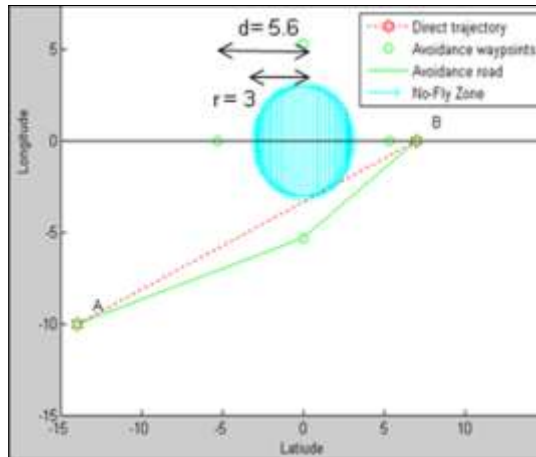


Figure 1. Optimized avoidance trajectory of NFZ between points A and B

Following the avoidance step of the NFZ the program accesses of a chain of waypoints. With the aim to create a viable trajectory for an airplane, algorithms of interpolation between the waypoints are used. Next, a meta-heuristic method that we created will be used to determine the shortest trajectory. This method combines the advantages of two already existing methods that we will present: the *descent method* and the *search method by taboos*.

The method of the descent [2] is an optimum local research method, where S is the assembly of the possible ways of avoidance and $f(s)$ is the length of the way of avoidance s .

In the program, there is the need to:

- 1) Choose randomly a solution s , a way of possible avoidances, in S
- 2) Determine randomly a solution itself s' , a way of possible avoidances, in S

There are two possibilities:

3.1) If $f(s') < f$ then $s = s'$

3.2) Otherwise return to the step 2)

Nevertheless, this method has a very constraining defect: it stops at the first found extreme. In other terms, it explores only a small part of the space of the solutions, and the first solution is obtained, acceptable accordingly to the criteria entered by the user. Other methods are further explored, such as the *search method by taboos*, and in order to use it, several steps are considered.

First of all, it is necessary to fix the criteria for stopping the optimization, such as the number of iterations. A reference solution is found and is fixed, and further, other solutions are tested. As soon as a tested solution is found to be better than the reference solution, and replaces it, then is deposited in a taboo list, which means that the other tested and non-withholding solutions are excluded from the next optimization research.

The minimum number of iterations is needed to obtain an optimum solution. If the stop criterion is the time, the number of iterations or the number of iterations without improvement, then the research becomes more efficient. The algorithm stops itself when the criterion of stop of optimization is attained.

The studied solutions are trajectories connecting the waypoints. The criterion to find the best solution is *the shortest trajectory is the best one*. Another criterion is further added, for the viability of the trajectory.

The solution retained by this algorithm resumes the principle of the descent method at which certain properties of the research by taboos was added. In fact, the program is provided with a criterion of stop of optimization that considers *10 successive iterations without any improvement*. This criterion, a compromise between the rapidity of the results and the effectiveness of the withheld solution, is found by experience during the program testing.

The list of solutions considered as taboos is at variable size: when a solution is added to the list of the taboos, it is added to the other solutions of the list. The created algorithm is therefore hybrid and innovative. It is further assumed that the solutions are obtained by polynomial interpolations, using the points present in the chain of waypoints.

These new avoidance points, situated between the waypoints, are obtained on the assembly of the trajectory following the avoidance of the NFZ. One of these functions is randomly chosen and its total length is calculated. This is the *reference function*. Next, the other tested functions are randomly chosen and measured in the same manner. If one of these functions is shorter than the reference function, it becomes the reference function. Then, the following parameters are defined:

- S is the assembly of the possible ways of avoidance
- $f(s)$ is the length of the way of avoidance s
- T is the list of the taboos and n a meter.

Then, the pseudo code of the developed solution for this program consists in the following steps:

- 1) To put $n = 0$ and $T = []$ (that means that the list is empty). Choose a solution s in S
- 2) To determine a solution s' in S
- 3) If $f(s') < f(s)$, then $s = s'$ and $n = 0$. Then, $T = [T, s]$: s becomes a solution taboo solution; this solution will not be again tested

4) If $f(s') \geq f(s)$, then $n = n + 1$.

There are next two possibilities:

5.1) If $n < 10$, return to the step 2

5.2) If $n \geq 10$, the reference solution is the final retained solution.

As shown in the legend of figure 2, the waypoints are represented by circles. The trajectory in full trait is the shortest existing trajectory, as it connects the consecutive waypoints by lines. All the stippled curves represent trajectories that were tested. The thick stippled trajectory is also a tested curve.

This shortest retained trajectory that is a straight line (already known in advance answer) was therefore founded with our algorithm.

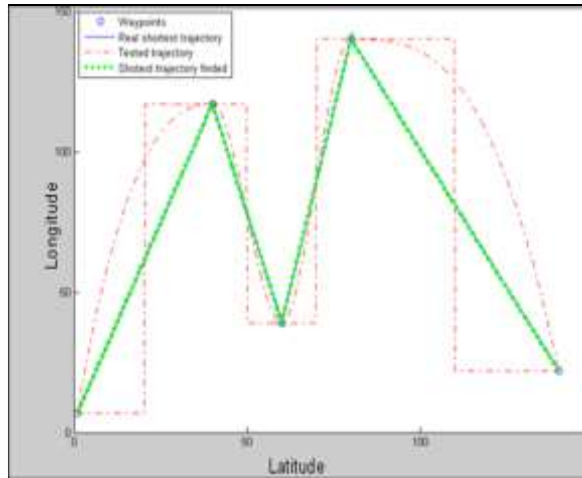


Figure 2. The shortest road found by combination between the taboo and descent methods

The available interpolations are the ones already programmed in Matlab: pChip, Cubic, Linear, Nearest, Spline [3]. To add another interpolation method, it is possible to add a Matlab function describing a new interpolation in the work file, and next to add it in the routine of random selection of the interpolation methods, in order to add it among the possible choices of these methods.

In order to calculate the limits in calculation time of the method that we created, several tests were realized. The goal is to find the time at which the calculation became too long. The following table shows the obtained results for different numbers of *waypoints*.

Number of points	Response time
2	Instantaneous
3	Instantaneous
5	< 0.5 sec
10	< 0.5 sec
15	< 0.5 sec
20	< 1 sec
25	< 1 sec
50	< 1.5 sec

We can note that algorithm is fast. It would be necessary to consider the calculations for the lines with 20 points and 25 points. In fact, during a transatlantic trip (that will be the example of our validated trajectory), the airline companies use approximately 10 waypoints. If one considers that the system avoids automatically a maximum of 4 zones, a number of 4 to 12 points of avoidance are generated.

It can be seen that the algorithm gives a fast response of less than 1 second in these situations. When the number of points approaches 50, the situation takes long time to be arranged. Nevertheless, this situation would correspond to more than 12 avoidances during the flight time, which is a high.

3. VALIDATION METHODOLOGIES

A number of hypotheses are established for the trajectories studies in four dimensions. The hypotheses are the following:

- All the waypoints are of the fly-over type
- The NFZ are considered fixed
- The roads of avoidance are free of all other NFZ
- The altitude changes are small
- The speed changes are small
- In the cruise regime, the speed changes are small, of the order 10%
- The airplane acceleration is constant

In this work, the resolution principle is tested, as well as the realism of the simulation. If the results are validated by considering a constant acceleration, a future work could be to replace this data by the plots of real acceleration (therefore a variable acceleration) to increase the realism of simulation.

At this stage, the system is functional in four dimensions. Thus, the developed algorithm, explained in details in the above section, is able to detect the NFZ on an initial trajectory, to propose the shortest possible avoidance road and to find the speed in every point of this trajectory.

To render the system more functional, and to allow its application on real cases, it was decided to use additional functions.

Software for the generation or validation of flight data will be created. The added functions calculate the following characteristics:

- The traversed distance in all points
- Calculation of the time of realized flight in all points
- The estimated hour of arrived time (Estimated Time Arrival, ETA)
- The fuel consumption
- The variation of the mass
- The necessary sprout force in all points

Each of these functions will be explained and its interest will be justified.

The method of length calculation concerns the discretization of the final trajectory in a very dense manner. Thus, the distance between two points is very small and the trajectory between two points is considered as a straight line.

This hypothesis verifies itself much better when the discretization is denser, and this is the reason why the sampling increased to more than 1.000 points between two consecutive waypoints.

In a trans-Atlantic flight, the distance between two waypoints always is always smaller (<) to 1,000 km, which signifies that the distance between two consecutive points of the sampled trajectory, is inferior to 1 km due to the sampling of 1,000 points.

The speed and the distance between each point being known, we calculate the flight time in all points, which is necessary for the Estimation of the Arrival Time EAT function.

The RTA respect is very important in the aeronautical exploitation, and this RTA function is incorporated in the current FMS.

In the report EEC Continuation n408 entitled *4D Trajectory Management Pilot Simulation*, written in June 2008 [1], EUROCONTROL describes the functioning of the FMS using the Estimated Arrival Time.

An error appears when the Estimated Time Arrival (ETA) goes out of the window predefined by the FMS of [-2 min; + 2 min] with respect to the RTA. The following figure, extracted of this same report [1], illustrates it.



Figure 3. Visual explanation of the alarm system RTA ERROR

On Figure 3, the left window shows a flight in which the ETA remains in the window of [-2 min; + 2 min] with respect to the RTA, more precisely the RTA is at 00:43:00 and the ETA is at 00:43:15. No alarms or warnings are released.

On the contrary, the right window shows a flight, and is seen that the ETA goes out of the tolerance window.

The RTA is always at 00:43:00, but the ETA is at 00:40:15. A warning RTA ERROR further appears. The system indicates also the estimated delay time with respect to the RTA.

In our computer program, the calculation of the ETA was set up (with the hour of departure as entry and the total time of foreseen flight) as well as a function of RTA_error that gives the error between the RTA and the ETA, and that precise if it is a matter of the advance or delay in the flight time.

It is important to calculate the fuel consumption, with the aim to optimize the flight in terms of cost and pollution. The internet site of Boeing [5] gives the following data:

- Maximum capacity of fuel quantity: 199.158 litres
- Ray of maximum action: 12,700 km
- Speed at 35.000 ft (11 km): Mach number of 0.84 corresponds to 933 km/hour

The estimations are done with the following formulas:

- Consumption (litre/km) = Maximum fuel / Ray of maximum action
- Consumption (litres/seconds) = Consumption (litres/km) * Speed (km/hour) / 3600

The calculation gives a kilometrical consumption of 15.6 litres / km, which is equivalent to a consumption of 4 litres / second. We further calculate the total consumption of the fuel, by knowledge of the flight time.

When the estimation of the fuel consumption is ended, the variation of the airplane mass is estimated.

The density of the kerosene at 35.000 ft has values between 0.755kg/litres and 0.840kg/litres. The reference considered for this work is the average of the two densities values, and is equal to 0.797 kg/litres.

At each point, the airplane mass is evaluated, while keeping in mind that the mass variation is due only to the fuel consumption.

The pilot concentrates on the speed and not on its conversion in sprout force. The most important parameters in the airplane flight are the following: the drag (in **N**), the thrust (**N**), the drag coefficient (non - dimensional C_x), the airplane surface (\mathbf{m}^2), speed (**km/h**), the air density at the cruise altitude $\rho(h)$ and the airplane angle of attack. The three cases to treat are: the cruise, the descent and the climbing flight regimes.

The waypoints were randomly chosen to represent real distances between them. The distances are expressed in kilometres (**km**), lengths in hours (**h**), speeds in kilometres per hour (**km / h**), accelerations in kilometres/hours/seconds (**km/h/s**), masses in kilograms (**kg**) or in tones (**T**) and fuel consumptions in litres (**l**).

The speed profile was carefully chosen: accelerated, decelerated and constant speed flights will be considered.

These parameters were chosen with the aim to test the automatic adaptation of the speed using real test data.

The latitude varies from 0 to 6,000 km, the longitude varies from 400 to 1,000 km and the altitude is constant at 11 km.

Please find positions of the waypoints: **A**(0, 400, 11), **B** (780, 530, 11), **C** (1332, 660, 11), **D** (1930, 800, 11), **E**(2630, 900, 11), **F**(3523, 1000, 11), **G** (4000, 970, 11), **H**(4750, 950, 11), **I**(5600, 940, 11) and **J**(6000, 934, 11).

The data inputs given by the operator are the acceleration profile, the required speeds at the waypoints, the fuel airplane consumption, the Departure Time and the Required Time of Arrival RTA.

In reality, certain data will be delivered by other embedded systems. The other values are automatically calculated by the computer program.

The length of the trajectory is 6061.7852 km
 Enter the plane acceleration profile (km/h/s): 35
 Enter the Waypoint Required Speed: 880
 Enter the Waypoint Required Speed: 880
 Enter the Waypoint Required Speed: 890
 Enter the Waypoint Required Speed: 890
 Enter the Waypoint Required Speed: 890
 Enter the Waypoint Required Speed: 890
 Enter the Waypoint Required Speed: 890
 Enter the Waypoint Required Speed: 890
 Enter the Waypoint Required Speed: 890
 Enter the Waypoint Required Speed: 880
 Enter the Waypoint Required Speed: 880

What is the airplane fuel consumption (l/s)? 4
 The total flight time is 6h 49min 45s
 Enter the departure hour: 12
 Enter the departure minute: 04
 The Estimated Time Arrival is 18h 53min 45s

Enter the Required Time Arrival hour: 18
Enter the Required Time Arrival minute: 55
RTA Error is 0h 1min 15s with advance
Do you want to save the flight data (Yes: press 1, No: press any other key)? 1
Data are available in **fly_data.xls**

The **fly_data.xls** records the flight test data of an aircraft for which each function is automatically calculated. The altitude is constant at 11 km.

This matrix records the distances (km), latitudes (km), longitudes (km), speeds (km/h), flight time (hours), remaining flight time (hours), fuel consumption (litres) and weight variation (kg).

The average speed is 887.63 km/h that gives a length of flight of 6 hours and 50 minutes, where time is coherent with the time proposed by the computer program.

The fuel consumption is 78.38 tones, that is coherent for a transatlantic flight. In fact, 98,000 fuel litres are used for a flight of 6,000 km. The ray of maximum action of the Boeing 747-200 is 12,700 km and its maximum fuel quantity is 199,158 litres. Once again, the coherence of results is shown: half of the maximum fuel quantity is needed to traverse the half of the maximum action ray. To validate the calculation of the thrust, we use the following entries:

The mass is of 370 tonnes (that will be converted in Newton units), the drag coefficient (C_x) of 0.075, the surface of 511m² and the air density obtained by the perfect gasses relationship in the troposphere: $\rho = p/RT$ where R is the constant of perfect gasses and equal to 287,1 J.mol⁻¹.K⁻¹, the pressure is expressed in Pascal at the chosen altitude and the temperature is given in Kelvin to the chosen altitude. For an altitude of 11000 m, the values are 22632 Pa for the pressure and 216.65 K for the temperature. The previous formula allows obtaining the air density of 0.3639 kg/m³.

The obtained values are situated between 416.670 N and 426.200 N for a cruise flight at a minimum speed of 880 km/h and maximum speed of 890 km/h. The Boeing 747-200 has four reactors with thrusts between 230 kN and 240 kN.

In order to validate our algorithm, a flight trajectory between Paris and Montreal will be simulated.

This trajectory will contain two problematic NFZ (that are located on the trajectory and it is possible to avoid them), and two non-problematic NFZ (that are not on the trajectory, and would not modify the road).

These zones are fictitious, but they can be considered as meteorological phenomena or aerial conflicts to be avoided.

A first non-problematic NFZ will be placed under the flight trajectory, in such way as if it was situated at the same altitude as the airplane, it could be avoided, and the other NFZ will be situated at the same altitude that the airplane, but not on the trajectory, which will show the robustness of the zones avoidances: the airplane would not have to avoid a "not bothering" zone.

Besides, the trajectory between two successive samples is considered as a line, the average distance between two points will be given in percentage with respect to the total length of the trajectory.

The altitude variations are also considered small. At last, the flight time will be compared to the real flight times found on the Internet sites of Air France [4] and Air Canada [5]. The real data in terms of burnt fuel will be compared to the Boeing data [3].

The coordinates of waypoints (abscissa, ordinate, altitude, passage speed) measured in km are the following: **A**(0, 375, 11, 900), **B** (428, 478,5, 11, 900), **C** (1500, 600, 11, 930), **D**

(2100, 600, 11,3, 933), **E** (3214, 600, 11,3, 933), **F** (4200, 500, 11,3, 933), **G** (5142, 357, 11,3, 910), **H**(5535, 214, 11, 900), **I**(5678, 142,8, 10,5, 900) and **J** (5857, 0, 10, 900).

The coordinates of the four NFZ (abscissa of the centre, ordinate of the centre, altitude of the centre, ray), given in *km* are the following: **NFZ₁**(964, 535, 11, 20), **NFZ₂**(2700, 600, 11,3, 10), **NFZ₃**(4800, 400, 5, 20) and **NFZ₄**(200, 25, 11, 10).

Validation of results

Please find the obtained trajectory in the altitude plane versus the longitude:

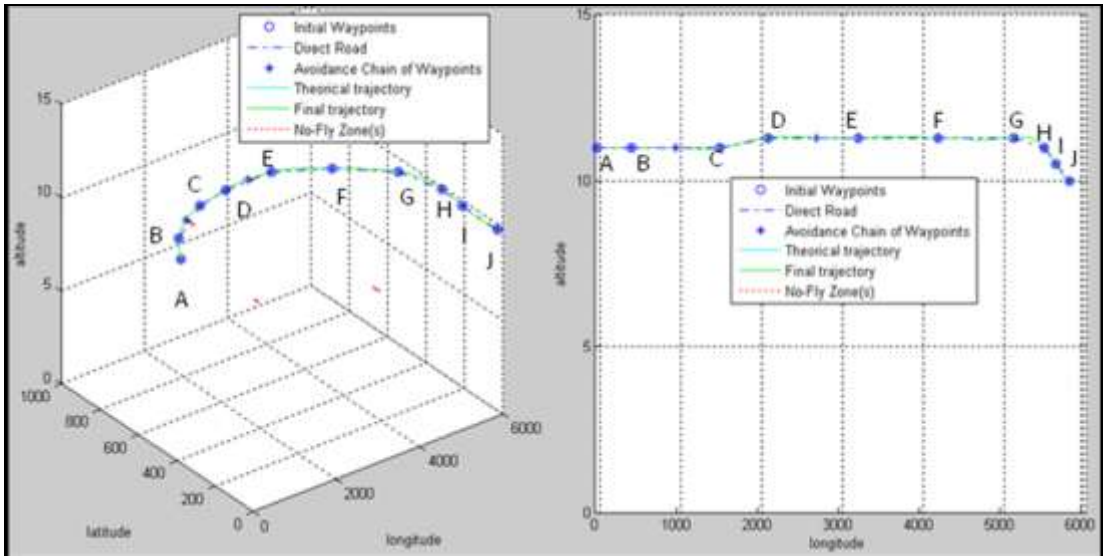


Figure 3. Trajectory in the altitude plane versus the latitude and longitude

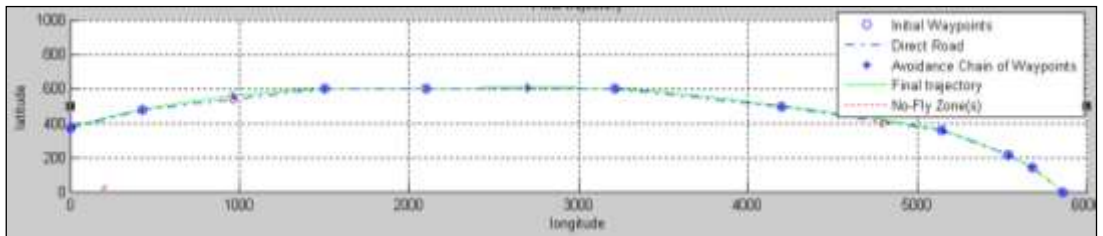


Figure 4. Trajectory in the latitude versus longitude plane

In the above schemes, the circled stars represent the waypoints of the initial flight plane. The stippled circles are the NFZ and the stars are the points of the waypoints chain that include the initial and the avoidance waypoints.

The stippled trajectory is the theoretical trajectory that connects all points by straight lines and passes through the NFZ. The trajectory in full traits is the shortest avoidance trajectory found by the algorithm.

The flight time length is of 6 hours 29 minutes and 39 seconds. The airline companies Air France and Air Canada sell the flight ticket for a flight length of 6h 45 minutes (AF344 takes off at 13h30 and lands at 15h15 with 6 hours of jet lag and AC871 takes off at 13h40 and lands at 15h20); this time corresponds to the entire time when the departure point is left until the time when the arrival point is attained.

Our airplane is modelled at the cruise speed between Paris and Montreal, nevertheless, the simulated time remains rather close of the real time to validate the airplane flight length.

The **fly_data** matrix, that records the flight data of the simulation in every point of the trajectory, contains 10,990 points to model a trajectory of 5.992,3 km that gives a smaller precision ($<$) for a point at all 550 meters. There is a point at every 0.01% of trajectory. The angles of variation being very small, the hypothesis according to which the trajectory between two points can be considered as a straight line is validated. It is the same for smaller variations of altitude ($<$) to 1 km at 0.01% of the trajectory.

Table 1. Results obtained for the weight variation, sprout force and fuel consumption

Distance (km)	Speed (km/h)	Fuel burnt (l)	Weight variation (kg)	Sprout force (N)
5976.09	900	93258.13	74326.73	439755.7
5976.34	900	93262.09	74329.89	439742.5
5976.59	900	93266.05	74333.04	439729.4
5976.83	900	93270.01	74336.19	439716.3
5977.08	900	93273.96	74339.35	439703.1
5977.33	900	93277.92	74342.51	439689.9

The **ETA** calculator is also correct as well as the **RTA_error** as seen on the next two examples:

Example 1

The total flight time is 6h 29min 39s

Enter the departure hour : 10

Enter the departure minute : 00

The Estimated Time Arrival ETA is 16h 29min 39s

Enter the Required Time Arrival RTA hour : 17

Enter the Required Time Arrival RTA minute : 22

RTA Error is 0h 52min 44s with advance

Example 2

The total flight time is 6h 29min 39s

Enter the departure hour : 10

Enter the departure minute : 00

The Estimated Time Arrival ETA is 16h 29min 39s

Enter the Required Time Arrival RTA hour : 15

Enter the Required Time Arrival RTA minute : 00

RTA Error is 1h 29min 16s with delay

The ETA calculation takes into account the time of departure and the flight time. It is also shown that the estimation of the RTA in both directions ($ETA < RTA$ or $ETA > RTA$) is validated.

4. CONCLUSIONS

In this paper, the steps for a software development that would optimize automatically the flight trajectories using the autonomous avoidance of the NFZ, is here shown. This software proposes the calculation of several flight test data (such as speed, the ETA or the fuel consumption) for the real time flights, the calculation of the cost index *CI* or the use of inputs to other airplane embedded systems.

The software was validated for the Boeing 747-200, but it could be used for any other airplane for which the necessary parameters are known.

The Matlab code is automatically convertible on other platforms. In fact, this code will be converted in C or in FLSim in order to be implemented on the flight simulators. Another new feature is the use of the new meta - heuristic method to solve the optimization problem.

Even if this method is a combination of two known methods, it offers an alternate methodology to the other existing methods, while combining the advantages of these two methods.

REFERENCES

- [1] *** EUROCONTROL EXPERIMENTAL CENTER. 2008. *4D Trajectory management pilot simulation*, EEC Report no 408.
- [2] *** The Matworks Inc., *The Matlab User's Guide*.
- [3] *** The Boeing Company, 2010, « *Technical Characteristics -- Boeing 747-400* ». The Boeing Company, online, consulted 20th of November 2009.
- [4] *** Air France, 2010, « *Réservations en ligne* ». Air France, on line, consulted 20th of November 2009.
- [5] *** Air Canada, 2010. « *Réservations en ligne* ». Air Canada, online, <www.aircanada.com>. Consulté le 20 Novembre 2009