Trajectory generation in a 3D flight plan with obstacle avoidance for an airborne launch craft

Svetlana Dicheva, Yasmina Bestaoui

Trajectory generation in a 3D flight plan with obstacle avoidance for an Airborne Launch Craft

Svetlana DICHEVA* and Yasmina BESTAOUI*
*Laboratoire IBISC
Université d'Evry-Val d'Essonne
38, Rue du Pelvoux, 91025 Evry, France

Abstract

For mission safety and efficiency, the airborne launch vehicle has to find an appropriate route to reach the mission goal satisfying some environment requirements and system constraints. A modified 3D waypoints generation path based on an improved version of the A* algorithm is proposed to find an optimal flight plan solution. This simple geometric path planning procedure can be implemented in real time in order to plan a new reference trajectory. In a presence of detected obstacles such as turbulence zones, no-fly zones, storms, etc., the information about the environment is regularly updated. This route reaches one or multiple goal points important for the mission success in order of their priority. The improved A* capabilities are tested via simulations in different scenario.

1. Introduction

Our studies concern an airborne launcher concept using an airplane carrier [1]. The innovating concept of airborne launching is currently studied in The French National Center of Spatial Studies (CNES); we are associated in this research within the project PERSEUS. The aircraft, the first stage of the launching system, transfers energy to the launcher. The launcher is not activated on the ground but is initially carried during the first part of the mission by an autonomous aircraft. At a certain given altitude, the launcher is released then is activated. The airborne launch proposes a better performance compared to a classical one. Moreover, the carrier would be reusable, reducing the launching costs.

A flight plan is necessary for the success of this mission. It is defined as the ordered set of movements executed by the system. It can be decomposed in phases. Each phase is described by the coordinates of waypoints associated with the system constraints at which the aircraft has to fly between these waypoints. A phase is completed when the final subgoal waypoint is reached. The airborne launch has to ensure divergent trajectories between the two stages at the separation as the carrier has to move away as quickly as possible from the launcher [2]. The airborne launch mission stages are presented on Fig.1. The mission planner allows the aircraft to reach the mission goals by taking into account the risks and obstacles which could appear during the mission. Obstacles could be bad weather conditions or non-fly zones. Decisional levels are necessary when the vehicle moves in a real environment. The two stages structure composed by a carrier and a launcher has several levels of interaction with the piloting systems [3]. The application of an automatic planner in the algorithms of guidance and navigation is a critical subject. All these levels represent a source of various constraints. The planner prepares the flight plan for a mission safety by generation of waypoints using planning algorithms. The most popular planning algorithms are potential field methods [4], Dijkstra and A* [5], Voronoï [6] and algorithms based on the object-oriented programming [7]. Dijkstra algorithm evaluates the cost to move from one node to any other successor node and gives the waypoints cost of the shortest path as the connecting cost between every two nodes lying on the itinerary. Another research is "Best-First-Search". This algorithm estimates the distance between the current position of the aircraft system and the mission goal.

To satisfy all these requirements, the presented 3D A* algorithm combines the advantages of Dijkstra algorithm and "Best-First-Search" algorithm. The modified 3D A* algorithm is not only designed to find the shortest path for each mission phase, but also chooses the best initial step to start the path, the first node is chosen in the line of sight from the source to the final destination. The trajectory waypoints are generated with respect of obstacle avoidance to preserve the launch carrier from damages or undesired behavior [8, 9]. These waypoint routes are applied in dynamic plan with periodical update of the environmental information considering the time of detecting obstacles and are suitable for any airplane kinematic constraints. Generally the mission planning algorithms could be treated in two ways: as a part of the path-planning problem where the obstacle avoidance is considered, and the generation of kinodynamical maneuvers. The path-planning problems with obstacle avoidance find a path that connects an initial, start configuration with a final, goal configuration. The airborne launcher has to be able to track this path. That is the reason that the planner is a part of the control loop.
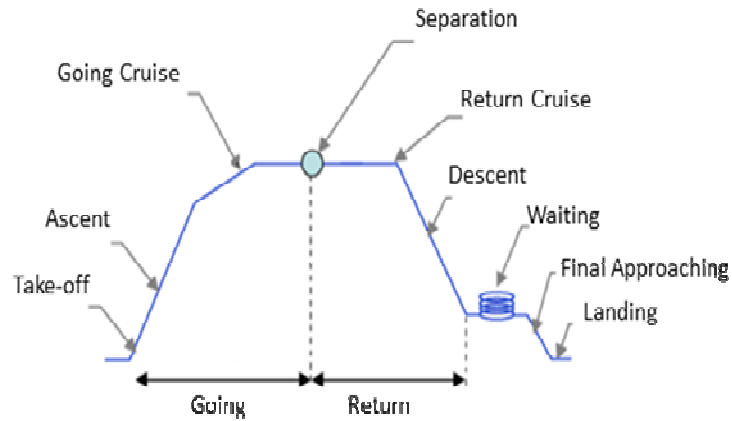
Figure 1: Mission stages

Off-line path planning is generated by 3D A* algorithm global planner. The path computed is often refined to ensure the path optimality. This operation needs high computational time and resource and cannot be done online. But global planners can be made suitable to be implemented online for path replanning in dynamic environments due to feasible demand of computational resources and time. While following the already generated path the sensor onboard systems transfer the updated information about the environment to the planning algorithm. When obstacles are detected the algorithms has to be able to compute new maneuvers. In case of replanning a path defined by waypoints to avoid the recently detected obstacles the new path also has to reach the goal point. This is called planning waypoint navigation. In [10] waypoint navigation system introduced in a database from the current information of the environment is composed by three main units: Reactive Unit, Knowledge Base and Deliberative Units. The Reactive Unit is the primary navigation when the vehicle is placed in the unknown environment. The information from the sensors is stored in the Knowledge Base. The already available waypoints are recorded over Deliberative unit for preplanning, planning or replanning.

The information for path cost, the distance to the next point and the orientation of the launch craft is needed. The next point can be determined by the coordinates of the previous waypoint, the heading and the time travel [11]. The path is generated in the coordinate system of the earth surface inertial reference frame.

This paper is organized as follows: Section 2 introduces the path replanning Problem Formulation, Section 3 presents the A* Global graph search planner. The Simulation results with different scenario with trajectory generation based on Cartesian polynomials are in Section 4. Section 5 concludes with some perspectives.

2. Problem formulation

The main problem for airborne launch mission safety is the avoidance of bad weather conditions as clouds, turbulence and ocean storms. It is supposed that this kind of launching missions will be executed in selected transoceanic routes where these phenomena are often seen. Recently NCAR (U.S. National Center of Atmospheric Research) developed a system that combines satellite data and computer weather models to predict involving storms or potential areas of turbulence. It is well known that the turbulence is the leading cause of injuries in all kind of missions in the aviation. Currently the human pilots have little weather information as they fly over the remote regions in the ocean, which is where some of the worst turbulence occurs. NCAR has worked a clear air turbulence global map based on global computer weather models with included winds and other atmospheric parameters. The areas of possible turbulence and higher clouds are drawn on satellite images; this information is sent to the aircraft and ground controllers with "up-to-the-minute" maps of turbulence. Some artificial intelligence techniques enable to forecast the movement and strength of the storm during next few hours. The on-board systems on the airborne launcher could use this data by a regularly update picture when it flies over the ocean.

The A* global planner is able to find the minimum distance path defined by the interconnections of waypoints to reach the goal(s) or subgoal(s) and accomplish the mission in a presence of different 3D weather models. The environment information is updated and the region with upcoming bad weather conditions could be predicted. The aircraft follows the already generated path until an environment changing appears then from a desired safety position the algorithm can replan the path to follow the same goal point(s) or to choose another sequence of goal points. With this flexibility the algorithm can be applied in different mission scenario with predicting moving obstacles.

In many references [10]-[11], to solve the problem for a shortest path for a system in a 3D space, the environment is transformed into a 2D model to use the A* classic algorithm. However, this could not be used for the transition from one configuration to another where the system intersects twice the same waypoint or waypoints with

the same coordinates for example on different altitude in the 3D search space. In the beginning, the search space has to be examined. In A* terminology, this space is called a map. The sum of maps costs helps the improved A* planning algorithm to find a safety route to the final point. In the presence of different kind of obstacles more route options exist between these two points. The A* global planner defines the waypoints from where the vehicle passes to reach the goal, another planner called local planner treats the moving styles to reach the subgoal(s) which could be the waypoints provided by the A* global planner. Cartesian polynomials, studied in Section 3, are used for this purpose. The A* global planner performs in 3D discretized map grids to find the correct path. These grids are defined by cells where the algorithm finds the path among nodes placed inside of the cells. The launch system kinematics is used to create the cell size and define the connections between the nodes so the path generated by these nodes is feasible. Obstacles even smaller than the grid size are also avoided successfully from the global planner.

Route planning precedes a decision to make for the best pathway configuration. The initial position is often a point nearby the take-off platform from where the vehicle will proceed automatically; the final position represents the launching or landing site or another intermediate position important for the mission success. Considering variable environment a replanning algorithm is necessary to solve the problems which cause variations in the nominal preplanned parameters. For this reason control levels help the aircraft to perform correctly during the mission, the A* algorithm implemented in the on-board system defines the future position and orientation of the aircraft with detected obstacles avoidance to produce a mission safety. The planner is a part of the control loop. A 3D A* planner represents a complex relation between the Guidance block and the Navigation used to manage the flight plan. The information for the new waypoint is based respectively on the position and velocity of the aircraft coming from the Navigation. This information could be upgraded regularly in interval λ and the planning of waypoints sequences gives the new data needed for the orientation of the aircraft following the new path. Once the system knows the path to be followed, this data is sent to the next level – Mission Control/Piloting. The Mission Control sends its commands to the Actuators which are operating on the airborne launch system. This interaction loop is closed by the data coming from the Sensors. The organization of the system control levels is presented in fig.2.
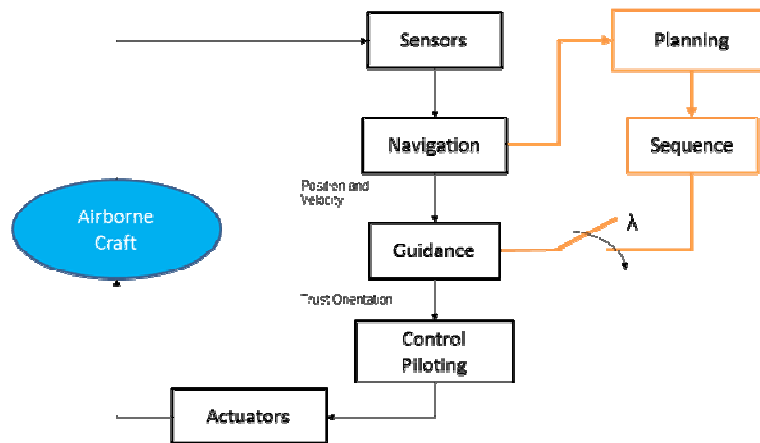

Fig.2 Mission control levels

3. A* global graph search planner

The region where the airborne launcher flies has to be defined. This region, the map, contains constraints called map costs. The planner uses these data to find the appropriated waypoints for mission safety. The cost maps are 3D grids, representing the area. Each grid cell could be of any different shape – square, triangular, hexagonal, it depends on the discretization of the region. In *XY*, *XZ*, and *YZ* coordinate systems in a function of how precisely the environment is introduced. The cells are defined by points in the space. The connection between the cells represents the possible waypoints where the aircraft could pass. The movements could be cell, edge or vertex movements. The cell movement represent the displacement in the near from one cell to every other cell in the neighborhood where the diagonal movements are also possible but with higher map cost. In the simulation results the center coordinates of each polygon grid are used to provide a reasonable set of nodes in the graph, the movement between the cells is called polygonal movement. The global planner makes a research within these nodes and finds the sequence of nodes with minimum map cost. The cost maps (maps with associated cost functions) are a geographical expression of the constraints of the desirable position of the vehicle. The sum of cost maps not only of different geometrical levels of the path but also from different constraints form the total cost map.

At first to find a sequence of nodes cell dimensions have to be defined, they depend on discretizition level of the grid. The grid is initialized by vertices with different length *dx*, *dy*, *dz* on *x*, *y* and *z*. In addition, the start and end

points are introduced. If the environment is with more complex obstacles with a high level of density on the flight region, it is possible to define a more precise map with small cells. Regions without obstacle could be composed by bigger cells. The 3A* planner is able to create a multilayer map according to the region evolution. The geometric search space has a rectangular shape. This space contains the initial position and the goal position. The parameters are defined: the maximum width, length and height of the region in the principal coordinate system as X_{max}, Y_{max} and Z_{max}. The dimension parameters need to satisfy some condition: the length has to be such that the horizontal coordinated of the initial and final point has to be inside the flight region. The height has to be bigger of the vertical coordinated of the initial or final point, and the width should be greater than the minimum turning radius of the vehicle, large enough to allow to sufficient lateral movement freedom. The ground is defined to be at Z_0.

Then the size of cells in the discrete cost map has to be defined: $\Delta X = X_{max}/S_x$, $\Delta Y = Y_{max}/S_y$ and $\Delta Z = Z_{max}/S_z$, where S_x, S_y and S_z are the range of the allowable connections between grid cells designed to capture the airborne craft kinematics. The cell range is planned as a function of R_{min} (minimum turning radius) and γ_{max} (maximum flight path angle) as presented in the formula below:

$$S_x = S_y = \frac{1}{\sqrt{2}} R_{min} \qquad\qquad S_z = S_x \arctan(\gamma_{max}) \qquad\qquad (1)$$

The advantage of the improved 3D A* algorithm is able to experiment with the different moving styles between the waypoints and gives solution for rich maneuvering fund. The A* algorithm's efficiency always was a much argued subject of scientific research because it consumes a lot of memory with the data of all extended nodes in the search space. This is necessary to find the optimal path by transverse all the grid cells. Of course this huge size of data could be optimized by itself as defining the most appropriated discretized search space. This space could be an air corridor in the mission airspace that the aircraft must remain in during its transit through the fly region. In last decades the efficiency has been more developed.

In many works, the 3D problems are solved with transformation into 2D space. This solution seems easy but in restricted area, as the underground railways or inner buildings, the overlapping layers appear frequently. In the airborne launching this could happen. Where the vehicle is taking off to accomplish the mission and back to the same site after, or during execution of complex maneuvers as full turning or cylindrical helices and etc. All manoeuvres are defined by the nodes coming from the improved 3D A* algorithm, the nodes are evaluated by the formula:

$$F(n) = G(n) + H(n) \qquad\qquad\qquad\qquad (2)$$

where $G(n)$ is the moving cost from the initial position to the current position and $H(n)$ is the estimated cost from the current position to the goal for each node n. The sum of the two costs defines the total moving cost from the start position to the goal.

In 3D space the current node does not have neighbours on the same level but now the moving styles increase with the layer above and below to the current plan. If in 2D plan there are eight moving styles on the left, right, forward and backward and also the diagonal movements between the mentioned ones, in 3D space there are 26 moving styles. The increasing number of moving styles gives more manoeuvring options corresponding to the airborne launch evolution in airspace. The algorithm cost maps can combine other cost values from different system requirements. The generation of successor points in the neighbourhood has to allow a sufficient freedom in the trajectory planning. The range of current state is an integer parameter named depth that represents the number of points in the neighbourhood from which successors are selected. In 3D space with twenty-six moving styles the depth is one, because there is only one point in each direction.

The A* algorithm classifies the successor points by their status, if they are obstacles which means rejected as a solution for optimal paths or they have been already found as solution. In the end of the algorithm this classification helps to reconstruct the optimal path from all parental nodes chosen as optimal waypoints. The planner finds the optimal path as a sequence of the interconnections between the waypoints. Often these interconnections are straight lines. They create sharp, not flyable routes with difficult maneuvers especially in 3D space. The two orientation angles γ (flight path angle) and χ (heading angle) of the aircraft carrier are obtained from the A* planner. The algorithm outputs are the location of the vehicle to reach the mission goal(s) or subgoal(s). To find the orientations the next formulation is applied:

$$\chi_i = a\tan\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \qquad \gamma_i = a\tan\frac{z_{i+1} - z_i}{(x_{i+1} - x_i)\cos\chi_i + (y_{i+1} - y_i)\sin\chi_i} \qquad (3)$$

Each configuration is associated with five parameters – three positions and two orientation angles. It is possible to introduce the time-to-go and velocity properties. These dynamic constraints are obtained in a function of the distance between two points. The time and velocity depend on the distance $D_{i,i+1}$ between two end points. The

distance is not the same in each direction; it is a function of every segment for vertical, horizontal or diagonal movement:

$$D_{i,i+1} = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \tag{4}$$

The average velocity V_a between the beginning and end of each configuration may be defined from the formula:

$$V_a = \left(\frac{V_{i+1} - V_i}{2} \right) \tag{5}$$

where V_c is the velocity of the system in the current node and V_p is the speed in the parental node. The airborne launch system has to satisfy the maximum and minimum velocity values $v \in [v_{\min}, v_{\max}]$ for each configuration. Then the flight time is:

$$t_{cp} = \frac{Dist}{V_a} \tag{6}$$

Both successive configurations $(x_i, y_i, z_i, \chi_i, \gamma_i, V_i)$ and $(x_{i+1}, y_{i+1}, z_{i+1}, \chi_{i+1}, \gamma_{i+1}, V_{i+1})$ can be joined by many interpolation schemes. Third order Cartesian polynomials are chosen because of their simplicity in real-time implementation.

$$x(s) = s^3 x_{i+1} - (s-1)^3 x_i + \alpha_x s^2 (s-1) + \beta_x s(s-1)^2$$
$$y(s) = s^3 y_{i+1} - (s-1)^3 y_i + \alpha_y s^2 (s-1) + \beta_y s(s-1)^2 \tag{7}$$
$$z(s) = s^3 z_{i+1} - (s-1)^3 z_i + \alpha_z s^2 (s-1) + \beta_z s(s-1)^2$$

where (x_i, y_i, z_i) is the ith way-point with associated velocity V_i and $(x_{i+1}, y_{i+1}, z_{i+1})$ is the (i+1)th way-point with velocity V_{i+1}. The coefficients α_x, α_y, α_z and β_x, β_y, β_z are kinodynamic relations, function of the time T necessary to traverse the segment:

$$\alpha_x = TV_{i+1} \cos(\gamma_{i+1}) \cos(\chi_{i+1}) - 3x_{i+1} \qquad \beta_x = TV_i \cos(\gamma_i) \cos(\chi_i) + 3x_i$$
$$\alpha_y = TV_{i+1} \cos(\gamma_{i+1}) \sin(\chi_{i+1}) - 3y_{i+1} \qquad \beta_y = TV_i \cos(\gamma_i) \cos(\chi_i) + 3y_i$$
$$\alpha_z = TV_{i+1} \sin(\gamma_{i+1}) - 3z_{i+1} \qquad\qquad \beta_z = TV_i \cos(\gamma_i) + 3z_i \tag{8}$$

The predicted arrival time T is approximated via eq. (6) using an average velocity. However, a detailed study has to be made in order to compute the minimal time respecting the kinematic and dynamic constraints of the system. The velocity continuity is assured. The velocity envelope could be found in the three directions, thus the orientations by the next formulation:

$$V(s) = \sqrt{\dot{x}^2(s) + \dot{y}^2(s) + \dot{z}^2(s)} \quad \gamma(s) = \arcsin\left(\frac{\dot{z}(s)}{V(s)} \right) \quad \chi(s) = \arctan\left(\frac{\dot{y}(s)}{\dot{x}(s)} \right) \tag{9}$$

An analysis in the curvature and torsion limitation between each pair of waypoints in a function of time transforms the rough paths into the smooth ones as a function of the abscissa s on x, y and z. The curvature is studied from the Frenet-Serret curvature-torsion model obtained by the relation with the orientation angles:

$$\kappa(s) = \frac{1}{\sqrt{\dot{\gamma}^2(s) + \dot{\chi}^2(s) \cos^2(\dot{\gamma}(s))}} \tag{10}$$

Once the path and waypoint based trajectory is generated, an observation of the time calculation and recalculation on in a dynamic environment has to be concerned.

The Guidance and Navigation module constantly estimates the aircraft position and velocity to be taken as inputs for the Control-Piloting block. With this information this decision making module is able to update in real-time. The quality of analysing the environment represents an important part of the first step in the obstacle avoidance in dynamic environment, where the autonomous system has to be able to track a safe route. Another option is to use some information of the path wherever there is no obstacle and recalculate the path from the preceding safe node to the goal node, slicing this intermediate path with the older parts of the path. This is presented in the simulation results. The improved algorithm could use old data from a previous 3D A* generated path and from an intermediate safe location it is able to evaluate a new path in a coincidence with the environment changing.

4. Simulation Results

Different numerical results are now presented to illustrate the applications of the path generation for an airborne launch vehicle. First, the environment is defined where the aircraft accomplishes the mission with specific coordinates in the map where the initial position and all the necessary goal points are initialized.

4.1 Feasible path replanning with a mobile obstacle

To present the feasible path generation the following parameters are chosen: $R_{min}=50\ m$ and $\gamma_{max}= +/-\ 30deg$ to define the translational range for each segment. The longitudinal, lateral and vertical dimensions of the geometric search space are selected to be $X=1000\ km$, $Y=1000\ km$, $Z=10\ km$ representing the environment where the carrier accomplishes the mission. The bad weather conditions or other no-flyable zones are represented by basic geometric forms and their combinations. The next step after the initialization of the environment is to place the initial position of the aircraft $x_{in}=57km$, $y_{in}=57km$, $z_{in}=0.3km$ (presented with the blue circle) and the final black star goal point with coordinates: $x_{in}=567km$, $y_{in}=567km$, $z_{in}=4km$. The non-flyable zone is defined with initial coordinates $O_1=\{169.7,\ 167.4,\ 0.74\}km$, and x, y, z dimensions as $\{113.5,\ 113.5,\ 1.47\}km$. The point obstacles (obstacles considered with small dimensions compared to the vehicle's dimensions) are $O_2=\{226.8,\ 339.3,\ 2.6\}km$ and $O_3=\{339.3,\ 339.3,\ 2.2\}km$ and a spherical cloud with its centre $O_4=\{452.5,\ 5452.5,\ 2.9\}km$ and its radius $R_{cloud}=1km$. Fig. 3.a shows that the proposed path search algorithm can avoid obstacles even smaller than the grid size and also generate the shortest path to reach the goal point. This path (with the blue line) is kinematically feasible with vertical and horizontal translational limitations. The simulation results are obtained via computations in Intel® Core Duo 3.00GHz. The goal point is detected *1s* after the beginning of algorithm calculations, the cloud is detected *2s* after the goal point and Obstacle1 on figure 7 is detected in *3s*. The total simulation time takes *51.65s*.

On already created rough path, the Cartesian polynomials are applied for each section to define a reference trajectory from the flight plan (fig.1). The scenario for the take-off, ascent and going cruise is studied where the obstacle avoidance will be necessary to preserve the aircraft. Second itinerary is generated based on the available data from the feasible path to create a smooth trajectory (green path in fig. 3.a.). For each phase a flight envelope is introduced with specific initial and final velocity.
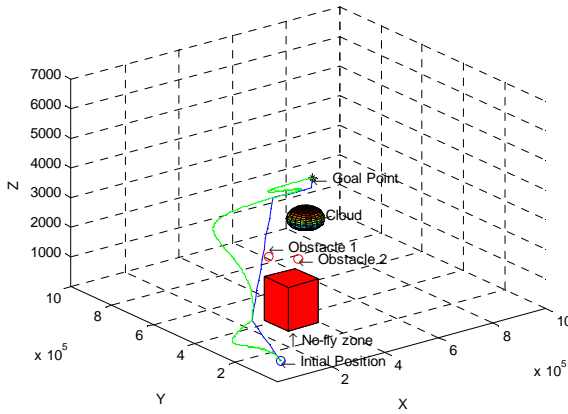

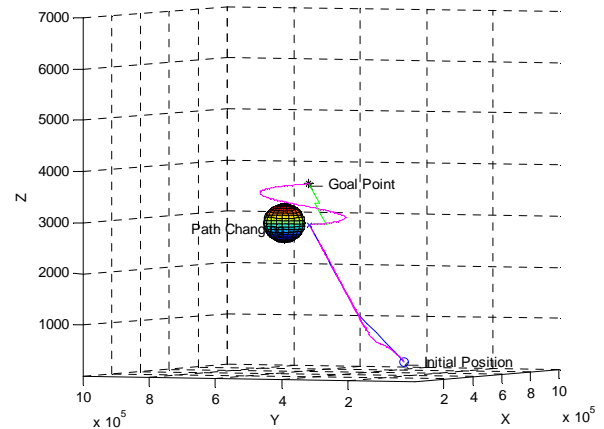Fig.3. a. Feasible pathway at the initialization   Fig.3.b Replanning of the path for the mobile obstacle

The next simulation (fig.3.b) shows the same scenario but this time the Cloud is detected on another position with coordinates $O_4=\{336.4,\ 524.9,\ 0.3\}km$. The airplane follows the already generated path in blue until one intermediate position from where the system detects a changing environment giving the order to the A* algorithm to prepare a new safe path presented on green in fig 3.b. Purple path represents the reference trajectory given to the guidance block. The coordinates of the cloud on the path changes are $\{311,\ 452.5,\ 0.29\}km$. This example treats another specification of the improved 3D A*, the algorithm could reach the goal using different initial position as intermediate position from previous paths. The cloud is detected in 3s from the beginning of calculation time for the new path. The total simulation time takes *3.04s*. As can be remarked, green path seems to be longer than necessary because of a too high predicted arrival time to join two successive configurations.

4.2 Multilayer grid with time and velocity definition

The next example (fig.4.) illustrates the trade-off between the length of each segment and the flight time from the current waypoint to the successive one. Each possible point in the search space is associated with a necessary

time to execute the translation between two points. This time initializes the time range in the same way as discretized space on *X*, *Y* and *Z*, the range being a reference to define the total flight time for the whole path in a function of desired speed. The velocity depends on the specific envelope to find the time-to-go. This new specification is applied in another discretized grid without obstacles and it is introduced in the existing one but with larger distance range to decrease the calculation time. In this grid a third and fourth goal point with coordinates: $P_3= \{866.7, 1800, 1133\}km$ and $P_4= \{1800, 1800, 100\}km$ are introduced. The simulation time for both paths until the 3th and 4th goal points takes only *2.23s*. It proves that in environment without many obstacles in a large distance, the algorithm is faster than in case with high density environment. The generated path has to be carefully chosen for aircraft safety. This efficient quality of the A* algorithm to introduce different constraints in one search space and in the same time to reach multiple goal points is an interesting advantage in the path planning. Once the length of each segment has been found in the same time an interval ΔS is associated to this segment to give more accurate description of the translation between the waypoints. This time is generated considering the interval to reach the second point of the segment in accordance to the total time to reach the goal itself. For the distance calculated from two points connected by two adjacent horizontal grid cells, the initial time interval is *Δt=5s*, which represents the time for a movement on the whole segment in the horizontal plan. To move from one horizontal plan to one vertical one [8, 9], this initial interval of time will change in a function of the length to reach the desired velocity. For example the time is more significant to move along the diagonal segments where the cost is higher. The constraints describing each waypoint are now with five segments: coordinates on *x*, *y* and *z*, specific arrival time to obtain a constant velocity and the velocity itself. The two cost maps in the search space function with different constraints.

1. the first one – the classical one with precisely cost map because of the complex environment and the new one with associated arrival time for each segment in a function of the desired velocity envelope. These two different maps executed in one search space define a new level of interaction between the grids transformed into multilayer grid.
2. The generated feasible path is able to avoid obstacles with different shapes and to reach multiple goals. The algorithm interconnects successfully different paths and it is able to begin its calculation at the moment where the last configuration was reached. This quality helps us to connect multiple scenarios in the search space and to generate a path in different complex situation.
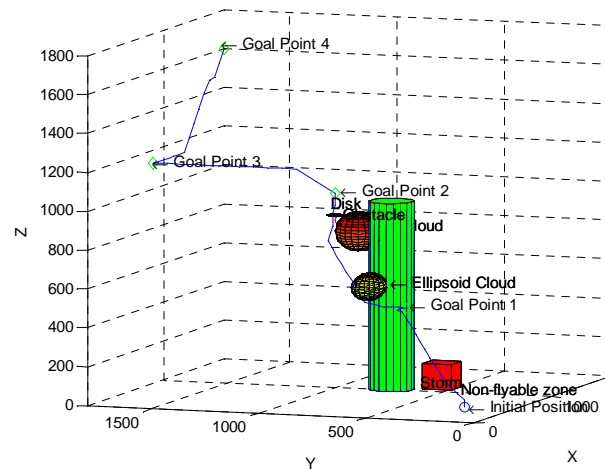

Fig.4 Optimal path, avoiding obstacles with four goal points in multilayer grid size map

V. Conclusions

This paper presents an updated flight plan based on a modified 3D A* algorithm, designed to reach one or multiple mission objectives defined by their priority, location and properties. Apart from producing a sequence of waypoints, the path planner designs a path that does not violate physical constraints of the vehicle. The algorithm gives the best shortest optimal path avoiding static or moving obstacles of different kinds. Several constraints are satisfied: the time necessary to accomplish the mission, the obstacle avoidance, and the dynamic constraints such as limited heading rate and flight path angle. The rough route is smoothed via Cartesian polynomials. The trajectory is configured such that the airborne craft passes within a specific distance of the waypoint with specific transition time between the path segments.

Analysis of simulation results of this improved A* algorithm shows that it is adapted to different environments where more flight constraints can be considered and that it proposes an optimal solution to generate an free obstacle path defined by a selection of successive trajectory configurations. With the method presented in this paper, new trajectory plans can be generated when knowledge of the environment is updated. In autonomous aerospace vehicle flight, the uncertainty of the environment is an important parameter for the practical application of this planning algorithm.

Our future work involves the introduction of the dynamic limitations of propulsion in the planning system and integrating the planning and trajectory generation algorithms into the control system. We are also interested in analysing the effects of the period of updates on mission performance, depending on the aircraft type and weather conditions. Then the transitory phenomena happening at the separation of the two stages and their influence on the proposed control method will be investigated. Finally, the real-time application needs to be studied and developed.

References

[1] N. Bérend, M. Bourgaie, S. Defoort, J. Hermetz, P. Schmollgruber, "Lancement aéroporté par véhicule polyvalent" ONERA report, Sept. 2008

[2] W. Oomkens, M. Mulder, M. M. Van Paassen, M. H. J. Amelink, "UAVs as Aviators: Environment Skills Capability For UAVs", IEEE, 2008

[3] D. Eppstein. "Finding the *k*-shortest paths", SIAM Journal of Computing, vol. 28, 1999

[4] F. Fahimi 'Autonomous robots, modeling, path planning and control' Springer 2009

[5] S. Gualandi, B. Tranchero. "Concurrent constraint programming-based path planning for uninhabited air vehicles", Proc. of SPIE, September 2004

[6] Mangiacasale, L., "Flight Mechanics of a μ-Airplane with a MATLAB Simulink Helper", Edizioni Libreria CLUP, Milan, 1998.

[7] E. Chanthery, "Planification de mission pour un véhicule autonome", PhD thesis, Univ. de Toulouse (France), 2006

[8] Y. Bestaoui, S. Dicheva, "3D Flight Plan for an Autonomous Aircraft", 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, January 2010, Orlando, Florida

[9] S. Dicheva, Y. Bestaoui, "Route planning for an autonomous aircraft", 49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace ExpositionJanuary 2011, Orlando, Florida

[10] M. Ibrahim, S. Ragavan, S.G. Ponnambalam, "Way Point Based Deliberative Path Planner for Navigation", IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, Singapore, July 14-17, 2009

[11] D. Mulvaney, Y. Wang, I. Sillitoe "Way-point based mobile robot navigation", 6th cong. on Intelligent Control and Auto., Dalian, China, 2006