



**HAL**  
open science

## Autonomous Detect & Avoid

Boubekeur Begue, Nicolas Capdeville, Jean-François Lamaudiere, Nicolas Senequier

► **To cite this version:**

Boubekeur Begue, Nicolas Capdeville, Jean-François Lamaudiere, Nicolas Senequier. Autonomous Detect & Avoid. ERTS 2018, Jan 2018, Toulouse, France. hal-02156489

**HAL Id: hal-02156489**

**<https://hal.science/hal-02156489>**

Submitted on 18 Jun 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Autonomous Detect & Avoid

## **Category:**

Critical Transportation Systems

## **Name and affiliation of authors:**

- **BEGUE Boubekour**, R&D Engineer, AKKA, Research AKKA Technologies Group.
- **CAPDEVILLE Nicolas**, R&D Engineer, AKKA, Research AKKA Technologies Group.
- **LAMAUDIÈRE Jean-François**, Work Package Leader AKKA, Research, AKKA Technologies Group.
- **SENEQUIER Nicolas**, Project manager, AKKA Research, AKKA Technologies group

## **Keywords:**

Airship, Avoidance, Detection, Drone, Obstacles, ORM, Quadrotor, Simulation, Trajectory tracking, VFH, VO.

## **Summary:**

The research department of the firm AKKA technologies, named AKKA research, is working on the development of an Unmanned Aerial Vehicle (UAV) for civil aviation. This project is called OMEGA and in this abstract we are going to present the actual state of the development, particularly the “detect and avoid” part.

### **1. Scope & Context**

Along with 21<sup>st</sup> century digital transformation and the increasing autonomy capabilities in transportation systems (i.e. ADAS car functions), safety of the moving vehicles and operational costs are key drivers of systems architectures and development. This leads to the development of systems capable of handling complex unpredictable situations that may occur in real environment and previously managed by human cognitive intelligence. In the mobility domain (transport systems), one of the most significant challenge deals with the management of the situations variety in an unlimited environment where multiple constraints can interfere with the trajectory a transport system intends to execute. Our scope of work focuses in the aerial environment with a perspective of potential applications on all possible air mobility systems, from piloted aircraft, balloons or helicopters to drone (Unmanned aerial vehicle) achieving the final objective of increasing autonomy level of the aerial vehicle.

The function we develop would be useful in various applications from complete autonomy level – in particular for long-range drone where communication latency may induce reactivity delays in case of trajectory conflicts with an obstacle – to the support in pilot’s decision-making and situation awareness in piloted aircrafts. Hence, the “Detect & Avoid” function development is one of the most critical challenges faced by system designers to achieve robust autonomous system integration for real operations meanwhile not threatening and if possible increasing global safety assessment. In that frame, the system has to take into account the variety of the environment constraints, e.g. fixed or moving obstacles (building, bird...), collaborative or not (Anti-Collision Avoidance System equipped commercial aircraft or not-equipped hang-glider), traffic, ground relief or weather related,... in front of its operational capabilities (performance, physical behaviour, usecase and mission) to determine safe and reliable operational solutions.

### **2. State-of-the-art analysis**

**Context:** Prior art is composed by press articles, UAV manufacturer’s sites, FAA, EASA Publications, Symposiums, International conferences, Books, Scientific, IEEE, Universities studies, Students Thesis...). This prior art is abundant due to the exponential development of UAV and notably automatic navigation systems and associated detect and avoid function. The avoidance function R&D is necessary due to UAV fleet increase, civil flight safety, safety of people on-ground. Regarding Intellectual property policy and strategy, only patents are listed in this summary.

**Prior Art Search:** A first prior art search ended in 27/04/2016, listed 138 patents (US/EPO/CN) related to detect, avoidance or combined functions. Considering the fast evolution of such domain a new prior search was performed on October 2017 with the following key words: UAV, Unmanned, Autonomous, Navigation, Vehicle, Obstacle, Object, Detect, Sense, Avoid, Avoidance, System, Methodology, Strategy, Algorithm, Switch, 3D, 4D,

Machine Learning, Fixed Wing, Simulation, Modelling. About 35 new patents were found from 2016 to 25 October 2017.

Actual tendencies: Regarding global prior art search, detect and avoid is actually a worldwide challenge. About half of the pertinent patents on the subject were filled from 2012 to 2017. The leading countries are United States, China and Europe. Many patents from Boeing, Thales, Sagem, Honeywell, L-3 Unmanned Systems, American Gnc Corporation, Raytheon, SZ DJI, IBM, Parrot, Amazon... But also these universities are active: MIT, Standford, Denver, North Dakota, Northwestern California Institute of Technology, Ecole Polytechnique Federale de Lausanne, Behiang, Nanjing, Tsinghua, Shandong (China).

Thematic Analysis: Regarding our concept, patents and associated claims can be categorized in the following domains:

Detection and Sensor: Detection and avoidance 3D by Millimeter-wave Radar, safety path selection and return to flight plan (CN 106950978 A / Filing date Mars 28, 2017), Autonomous avoidance using Lidar, sonar (CN 105955303 A / Filling date July 5, 2016). Method with distance and thermal (US2017275023 A1 / Filling date March 28, 2016), acoustic, visual, infrared, multispectral, hyperspectral, or object detectable signal emitted or reflected from an object (CA2968864 A1).

Avoidance systems and algorithms: Navigating UAV with obstacle algorithms selection (US 7228232 B2 / Filling Date Janv 24, 2005). Multi-obstacle avoidance using iterative algorithm (CN 106919181 A / Filing date Oct 20, 2016). Algorithm using a speed obstacle cone, multi obstacle (CN 106647812 A / Filing date Feb 15, 2017). High-precision obstacle avoiding method with laser and differential GPS (CN 104850134 A / Filing date: June 12, 2015). Algorithm taking account dynamic forces analysis (CN 107015571 A / Filed May, 12 2017). Improvement of Velocity Obstacle Arc method, classification on three levels of obstacle threats CN 106292712 A / Filing date: Oct 25, 2016), Improvement of Potentia Field method, virtual force computing (CN 107102650 A / Filing date May 27, 2017). Sense and Avoid function (US 20150277440 A1 / filling date March 25 2014). System with rerouting automatic courier UAV (CN 104834319 A / Filling date May 5th 2015). System using multiple radar array (US 8378881 B2 / Filling date Oct 18th 2010). Fusion of depth information and optical flow vector (CN106681353 A1), Negative-portion guiding (CN106647810 A).

UAV environment Modelling: Multi sensor use to build egospace representation (US 20170193830 A1 / Filing date Dec 29, 2016). Airborne widefield airspace imaging and monitoring (US 8494760 B2 / Filling date Dec 14, 2010). 3D environment modelling using velocity data of a vehicle (US 20170178352 A1 / Filling date Dec 19, 2016).

Obstacles path prediction, multi-UAV: Multi obstacles avoidance (fix or moving), multi flight plan (US 8543265 B2 / Filling date Oct 20, 2008). 4D-GIS based virtual reality for moving target prediction (US 8229163 B2 / Filling date Aug 22, 2008). Method to avoid other a/c by UAV (US 20100121574 A1 Fill date Sept 5 2006). Trigonometric avoidance method for moving obstacle (US 8700306 B2 Fill date Janv 4, 2013).

Algorithms selection, redundancy: Multi sensor system and many avoidance algorithm, selection depending of environment complexity and speed range (US 9625909 B2 / Filing date Apr 1, 2016). Multi sensors, a bank of algorithms access to estimated environment, (US 20160125746 A1 / Filing date Mar 4, 2015). Multi sensor (GPS, Inertial, Lidar, Ultra-wave, barometric), multiprocessor and avoidance strategies selection (US 9604723 B2 / Filling date / Apr 1, 2016). Evaluation of several algorithms and selection by weighted criteria, artificial intelligence (US 7769474 B2 / Filing date: Sept 20, 2005). Method with avoidance flight path selection (US 20150379876 A1 / Filling date June 25th 2014).

Simulators, Matlab and C++: Simulating an UAV and environment in which the UAV is flying (US 8942964 B2 / Filing date Jun 8, 2010), Thesis UAV Simulation Environment for Autonomous Flight Control Algorithms Ondřej Karas 2012.

Synthesis OMEGA "detect & avoid": Systems for detection already exist (Radar, Mmv, Lidar, CCD camera, Infrared, acoustic, sonar and the combination of many sensor to build environment also exist. Associated to this sensor, avoidance algorithm already exist (Potential Fields, Vector Field Histogram, Dynamic Window, Velocity Obstacle, Nearest Diagram Navigation, Obstacle Restriction Method, Optical Flow). According to Prior Art some patents improve these methods/algorithms and some methods combine several algorithms and select one depending on environment complexity, UAV dynamic forces or speed/configuration. As many Sense & Avoid solutions already exist (including with multiple algorithms), the main aspects that could be innovative is the methods used to improve the robustness and redundancy of the avoidance function in order to improve the global flight safety. The main areas where few patents / studies / research were found are (the more to the less):

- The combination of avoidance strategies/algorithms to cover environment complexity and fix, moving obstacles,
- The alternative use of algorithms depending on configuration, flight phase, speed, environment, sensors performance
- The algorithms switching laws depending on UAV & environment parameters and sensors availability/failures
- The avoidance strategies monitoring to detect anomaly or disparity of flight path and then to suppress it from the available flight path choice.
- Artificial Intelligence and Machine Learning, UAV will decide strategy to adopt in real time and so the associated algorithm and evaluate the criteria and parameters to decide to change algorithm. Depending of experience, being able to improve the algorithm / strategies.

### 3. Positioning with regards to the state of art

In this section we are going to present the methods used in our model to detect and avoid obstacles (fix and mobile).

The study starts with identification and characterization of all the potential methods that could be used for obstacle avoidance, whatever the domain they belong to. Main source comes obviously with robotics engineering but are more likely applicable on ground, which means 2D environment. Development furtherly deals with adaptation to 3D environment. Our objective was to implement several methods in the calculator in order to provide robust and safest avoidance trajectory with regard to the in-flight surrounding obstacles, position targeted and global dynamic behaviour of the whole system.

- Study of optimized trajectory determination

Algorithms constitute the best functional candidates to handle complex deterministic constraints that could be encountered during a real flight. At the basis, algorithms were developed to determine shortest trajectory path between a departure point and an arrival point for video gaming applications. Then, those algorithms were largely used since several years in the mobile robotic domain. Limitation resides mainly in the fact that only 2D environment is considered in most of those applications. Another key issue is that sometimes, with a large amount of obstacles, the algorithm may not issue to a trajectory solution and is operationally mitigated on ground by stopping the robot displacement. Indeed, this solution is not acceptable for an aerial vehicle because in most of cases, it cannot stop its flight in a stabilized position (except some particular capabilities that belong to rotative wings configurations). Thus a compromise has to be technically chosen between the various avoidance trajectories in order to:

- Safely reach the targeted position
- Take into account in real-time any change of the environment constraints (newly detected obstacle, obstacle unplanned displacement)
- Take into account aerial vehicle capabilities to execute the trajectory required by the avoidance method
- Identify and execute the best trajectory – meaning the shortest or lowest energy-demanding - among the various potential trajectories solutions

Finally, the study aims to assemble all the various avoidance methods and to operationally determine which one is more appropriate. Thus a predictor is implemented in the whole system, simulating at  $t-1$  the potential trajectories, calculating displacement cost and, in fine, defining optimized trajectory segments the vehicle will then follow at time  $t$ . This assembly contributes directly to the function's safety thanks to the introduction of a dissimilar functional architecture based on various independent algorithms that have their own arguments running and solution cropping.

On the following sections are described the potential avoidance methods principles implemented in the system.

- **Vector Field Histogram method (VFH)**

The Vector Field Histogram is a reactive method based on a probabilistic approach. First, an active zone is defined around the aerial vehicle and this zone is divided in multiple small sectors, like a grid. In each sector is determined how many times the obstacle has been detected in this sector and it fills the grid. Then a histogram is build based on the result of the grid, the user defines a threshold and the algorithm provides the optimal direction [1],[2].

- **Obstacle Restriction Method (ORM)**

The Obstacle Restriction Method is a method that uses the “divide and rule” motto. In fact the principle of this method is that the algorithm will calculate an intermediate step before the final step. There are two possible solutions to place the intermediate step: between two obstacles, if the obstacles are separated with a distance longer than the aerial vehicle diameter, or on the direction of the obstacle’s edge but with a distance longer than the drone diameter [3].

- **Velocity Obstacle (VO)**

The Velocity Obstacle method is a method that separates mobile and fix obstacles using the sensors. This method needs to know the position of each obstacle at the times t and also t-1 to predict its direction and its velocity at the times t+1. Then, knowing the velocity of the aerial vehicle and all the obstacles the algorithm defines a cone which leads to a collision for each time and each obstacles. At the same time, knowing the aerial vehicle’s dynamic, a rectangle named Reachable Avoidance Velocity is defined and it contains all the possible velocity for the vehicle. Finally the algorithm chooses a velocity which is inside the rectangle but outside all the cones [4].

#### 4. Modelling

A simulation approach (MATLAB/Simulink environment) is used in order to test digitally avoidance algorithms before flight tests. Flight simulator software has been created. It embeds flight path monitoring, reference trajectory generation and trajectory tracking for an airship model. Furthermore, obstacle detection based on multiple sensors has been integrated. Finally 3 avoidance algorithms (Vector Field Histogram, Obstacle Restriction Method and Velocity Obstacle) have been adapted to an aircraft evolving in a three-dimensional space.

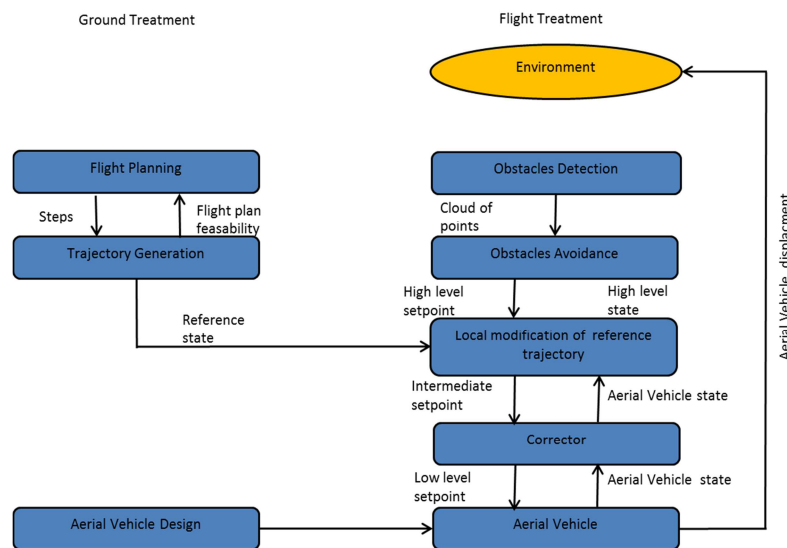


Figure 1 : System Architecture

This model is able to approach a real aircraft thanks to its dynamic constraint and its aeronautic constraint implemented inside. It contains 2 types of aircrafts: the first one is the airship which has similar behaviour than an aircraft which uses fixed wings but the final goal is to have a real plane model in order to have a realistic fix winged model. The second aircraft with rotative wings corresponds to a quadrotor aerial vehicle.

#### Feedback Control (low-level) and corrector

The system state is represented by the following state vector:

$$X = (x \ y \ z \ \gamma \ \chi \ V \ T \ \sigma \ \alpha)^T$$

With:

x, y, z - aircraft position

$\gamma, \chi, V$  - velocity vector (elevation, yaw, linear speed)

T,  $\sigma, \alpha$  - primitive of control vector (thrust, roll, incidence)

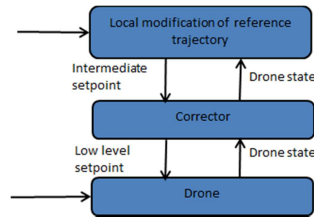
The control vector groups the physical quantities on which the corrector is able to modify the state of the system:

$$U = (\dot{T} \ \dot{\sigma} \ \dot{\alpha})^T$$

Where the components of the vector  $U$  correspond to the temporal variations of the thrust, the roll and the incidence.

In addition, a sampling period is defined for the needs of the simulation. At every time step the corrector calculates a new control to be applied and the system is updated.

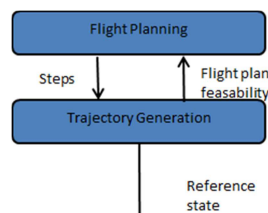
In order to generate an optimal feed-back control law, the used corrector is based on the non-linear control "Pointwise min-norm"[5].



This control is based on the theory of robust control of Lyapunov. In order to avoid a laborious setting, the system is linearized in one or several states of functioning. Finally the equation of Riccati is solved by means of the function `lqr` of Matlab (Control System Toolbox). Three weight matrices are introduced in order to determine the solution. The two first allow weighting the  $X$  state in front of  $U$  control and vice versa. After several iterations, it is possible to find the good adequacy between performance and robustness. Finally, the third matrix allows obtaining a compromise between the effort of control and the rate of convergence of the  $X$  state to the reference state.

### Trajectory generation

The generation of the reference state allows determining before the flight the instruction (procedure) to apply to the system in order to follow the defined flight plan. This instruction corresponds to the reference state  $X_{ref}$ . The limits and the constraints of the system are taken into account even before the take-off of the aircraft, because they validate or not the realization of the flight plan.



The generation of the instruction is made in three stages which allow reconstituting a reference state:

- 1- Spatial and temporal interpolation of the flight plan by cubic spline ( $C^1$ ).
- 2- Calculation of the velocity vector  $(\gamma, \chi, V)$
- 3- Calculation of the primitive of the control:  $T, \sigma$  and  $\alpha$  by inverse dynamics

The last calculation involves all the dynamics of which the aerodynamics of the aircraft. This resolution is based on inverse dynamics of the aircraft [5]. In these conditions, by setting in entrance  $(\gamma, \chi, V)^T$ , a block Simulink returns  $(T, \sigma, \alpha)^T$ .

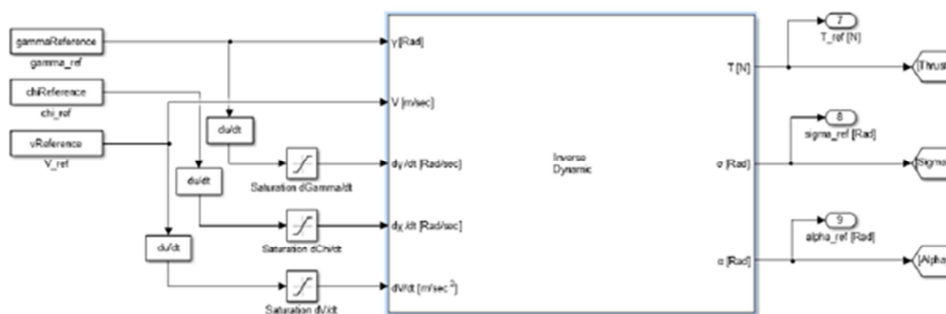


Figure 2 – Simulink bloc – Calculation of primitive for reference control

In this case, a solver with fixed step, the value of which is not equal to the sampling period, is used. This solution is chosen to respect the sampling time of the velocity vector. In most cases, convergence is satisfactory. In case

of too binding flight plan, he can have saturation and not convergence there. In these conditions, the use of solver with variable steps, more robust for the resolution of algebraic loops, is imperative.

### Detection function

Prerequisite for avoidance is detection, in other terms the input data for avoidance method are data issued from the detection.

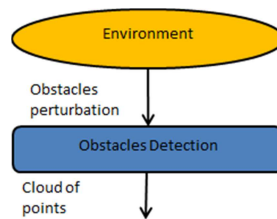
The aim is to have different obstacles detection technologies that have their own characteristics, which can be selected depending on the flight step and the current situation of the aircraft. Obstacle avoidance strategies are sensitive to the detection sensor used. Combining different sensors allows us to increase the detection capabilities of the aircraft by processing different types of data and combining them to have a more accurate representation. The simulation tool must be able to detect the different obstacles using these sensors. In this paper, two detection technologies are simulated, whose main characteristics are given below:

Sensor	Range (m)	$\Delta$ (m)	Range	Azimuth (°)	$\Delta$ (°)	Azimuth	Elevation (°)	$\Delta$ (°)	Elevation	Velocity
Ultrasonic	5	0,01	360	23	180	17	No			
LIDAR	100-150	0,03	360	0,3	30	2	Yes			

### Notes:

- $\Delta$  is the measure accuracy of the given data. Some sensors can estimate instantly the velocity of an obstacle and help for prediction of the obstacle's direction. With regard to the avoidance method, this method can be useful.
- Ultrasonic sensors are placed all around the aircraft, and we get spherical obstacle detection. Because of the short range of ultrasonic sensor, this kind of detection is useful during landing or take-off phases but not during high speed flying phases.

The Principle of avoidance function is given below:



- 1- The selected sensor creates a detection meshing, which depends on the sensor resolution and the detection area (range, azimuth and elevation)
- 2- An interpolation is made between the detection meshing and the detected obstacles
- 3- A point cloud representing the closest points of the obstacles to the aircraft is given by the simulator, according to the sampling period.

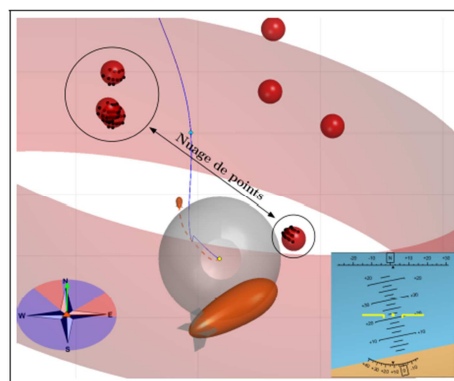


Figure 3 – Obstacles detection with a LIDAR sensor

Each detection point is named obstacle vector. The set of obstacle vectors establishes the cloud of points.

## Avoidance function

The determination of the avoidance trajectory uses a cost function (azimuth and elevation). The choice of the optimal direction is given by:

$$g(\delta_c) = \underbrace{\mu_1 \cdot \Delta(\delta_c, \delta_t)}_{\text{Orientation towards the target}} + \underbrace{\mu_2 \cdot \Delta(\delta_c, \delta_k) + \mu_3 \cdot \Delta(\delta_c, \delta_{k-1})}_{\text{Trajectory tracking}}$$

Orientation towards the target                      Trajectory tracking

$$\Delta(\delta_1, \delta_2) = \min(|\delta_1 - \delta_2|, |\delta_1 - \delta_2 - \Phi|, |\delta_1 - \delta_2 + \Phi|)$$

With:

$\delta_c$ : candidate direction

$\delta_t$ : direction towards the target

$\delta_k, \delta_{k-1}$ : current and previous direction

$\Phi$ : limit of the detection area

Candidate direction cost:

$$G(\delta_c) = [\mu_4 \quad \mu_5] \cdot g(\delta_c)$$

With:

$\mu_4, \mu_5$ : azimuth and elevation weights

## 5. Summary of major results

One of the key features of this model is that it is able to choose the best avoidance method during the simulation. In fact, some avoidance methods are more adapted to low velocities than high-speed velocities. It can also depend on the number of obstacles that the aerial vehicle will detect or on the total weight of the vehicle. To sum up the aircraft embedded function will choose the safer and the more direct and less expensive (in terms of displacement length) method to avoid the obstacle. In addition to that, the sensors that the model will use to detect the obstacle will also be adapted to the flying phase; the model gathers characteristics of multiple types of sensors such as ultrasound or Lidar.

We are going to focus on 3 key drivers of our design: the possibility to choose the method depending on the flight mode, the trajectory tracking and the different aircraft configurations that we can choose.

- Choice of the avoidance method

To help choosing the best strategy, these criteria must be considered:

- 1- Safety/Security : avoid collision by respecting a security distance between the aircraft and the obstacles
- 2- The flight dynamics : the avoidance method must be compatible with the real behavior of the aircraft
- 3- The optimal trajectory : control the distance between the theoretical and the real trajectories
- 4- The respect of the load limits
- 5- The respect of the restricted areas like airport, military bases, ...

Two approaches are possible:

For the first one, the choice of the avoidance method to be applied requires a phase of anticipation/prediction. The solution may be based on the use of a ghost aircraft, which is going to test several candidate avoidance trajectories in advance (ahead of the real flight) with regard to the "real" aircraft. In these conditions, the approach is the following one:

- Extrapolation of the position and velocity of the aircraft and obstacles from the previous data.
- Determination of the possible solutions, in term of avoidance, for all the next time steps. Indeed, it is thus required to store all the potential solutions obtained with all available methods.
- Implementation of the best obtained solution which respects the previous 5 criteria. Each of these criteria is weighted, in order to provide best results with regard to flight mission operational constraints (Aircraft configuration, navigation performance, allowed or prohibited fly zones, safety &/or cost operations drivers...etc.).

This first approach is currently under development.

The second approach is based on return of experience. For that, we perform a study on the flight parameters: aircraft behaviour (velocity, dynamics, distance to the obstacles, obstacles detected), number of obstacles and get the main parameters that will help to choose the best strategy. A database, containing different types of



obstacles configurations, is created. Each type must be ranked according the obstacle configuration complexity. The test defined previously helped to the database creation.

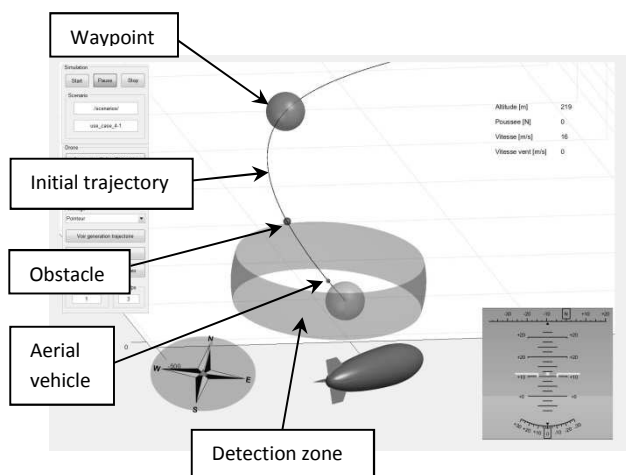
For example, the VO method is more adapted for high velocities and the vehicle will use this method if it encounters obstacles during its fast flight segment. If the velocity is lower the vehicle will use the ORM or the VFH method pending others criterions presented upper like the amount of obstacles, the total weight of the aircraft or its actual position to determine which one of ORM and VFH method is more adapted and leads to best trajectory optimization.

When the method is chosen the algorithm create an intermediate step which allows the aerial vehicle to continue its motion without collision with the obstacle and a new trajectory is created at each new step. We will present the trajectory tracking in the following paragraph.

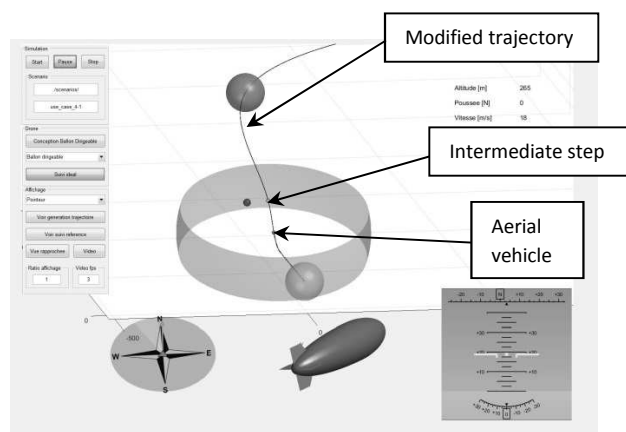
- Trajectory Tracking

The motion of the vehicle is defined by a trajectory that it has to comply with. This trajectory is generated by the algorithm and joins all the waypoints, and if an intermediate step is generated by one of the avoidance method the trajectory is modified and will join all the waypoints and the intermediate step.

The trajectory is defined considering the physical limitations and constraints of the aircraft as well as the required time delay to reach each waypoint. Here is an example presented on our human machine interface of the simulator:



- *Figure 4: initial trajectory*



- *Figure 5: modified trajectory*

The third important point in that model is that there are two types of vehicle flying behaviour implemented inside, the rotative wings and the fixed wings.

Actually, the fixed wings model is an airship model that can be comparable to a plane model but the final objective of the project is to develop a plane model to have a realistic fixed wings aircraft model. The rotative wings model corresponds to a quadrotor drone. The user will choose the model that he wants directly on the human machine interface and after the selection the model with associated dynamic constraints and limitations will change.

## 6. Validation

In order to validate our model/approach, a set of test cases was set up (about 50). These tests are classified in several categories: functional test, robustness test and operational test (use case).

So, a particular tool was developed to generate different scenarios. The input data are: a flight plan (waypoints set), its physical characteristics (dimensions, weight, inertia...), performance data (velocity [min and max], acceleration [min and max], admissible load factor...) and obstacle data (dimensions, weight, velocity and trajectory that can be straight, circular or random). Proposed scenarios can be :

- the aircraft can meet stationary obstacles (e.g. electrical cable) on its trajectory or near strategic points like waypoints, landing or take-off areas
- moving obstacles (e.g. birds or another aerial vehicle) going faster in the same or different direction that can cross the initially planned trajectory of the aircraft.

In order to prepare the tests in real configuration of the obstacle avoidance function described below, we have

implemented an ADS-B air traffic reception station. So, the data concerning position, heading, altitude, speed, airway, type of aircraft collected by the receiving station are transmitted to the ground control station and come to enrich the Human Machine Interface by showing the position of aircrafts detected on the background of vectorial map. The next stage will consist in confronting virtually the algorithms of the avoidance function with this real traffic and thus estimating the reliability of these algorithms in representative situations. The scenario presented in this paper is a use case which combines fixed and mobile obstacles (rectilinear and random trajectories). Data of the flight plan are given below:

	Time(s)	X (m)	Y (m)	Z (m)
<b>Departure</b>	0	0	0	200
<b>Waypoint 1</b>	90	0	1000	500
<b>Waypoint 2</b>	180	1000	2000	500
<b>Arrival</b>	360	0	4000	200

The used sensor is a LIDAR (range 150 m). The function trajectory generation gives a path of 4880 meters length. Here is an overview of the human machine interface:

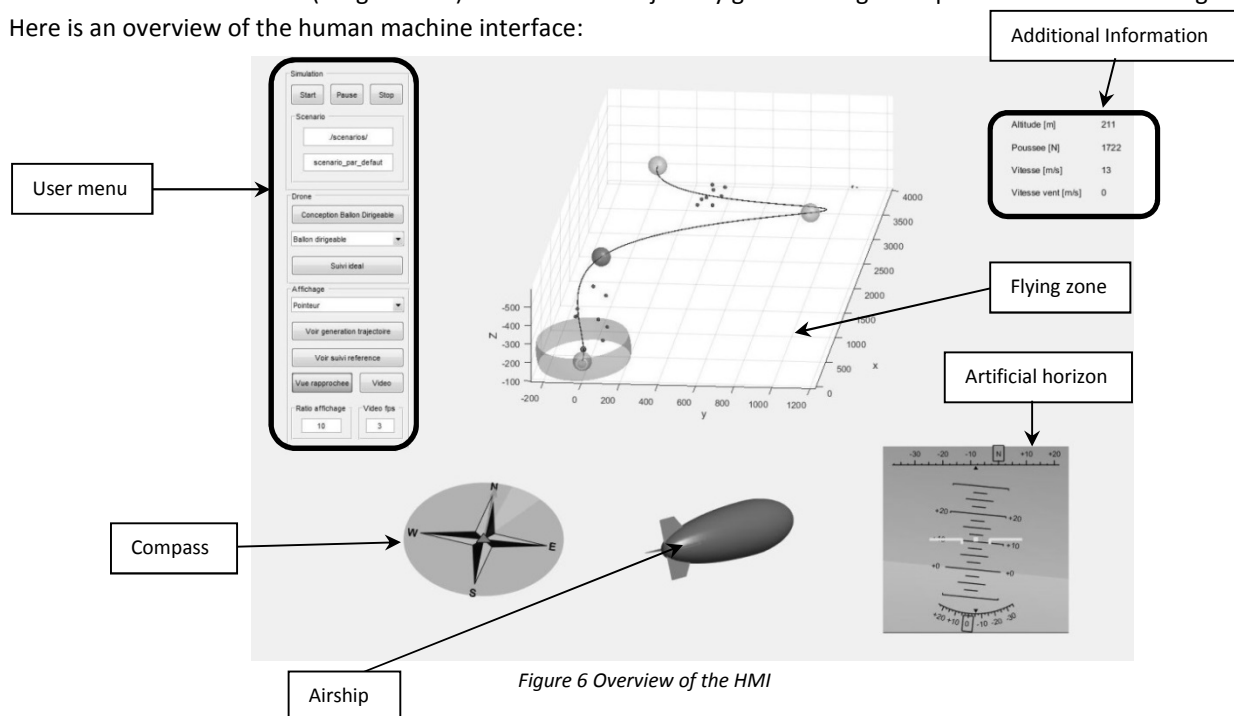


Figure 6 Overview of the HMI

## 7. Results and analysis

The final path length, with avoidance manoeuvre, is equal to 5107 meters (initial length 4883 meters), which represent less than 5% of distance additional cost.

The table below gives statistical results, with the use of velocity criterion only:

	ORM	VFH	VO	Total
%	27.12	4.08	68.80	100.00

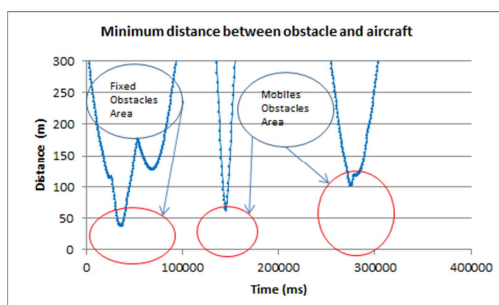


Figure 7 – Distance between obstacle and aircraft

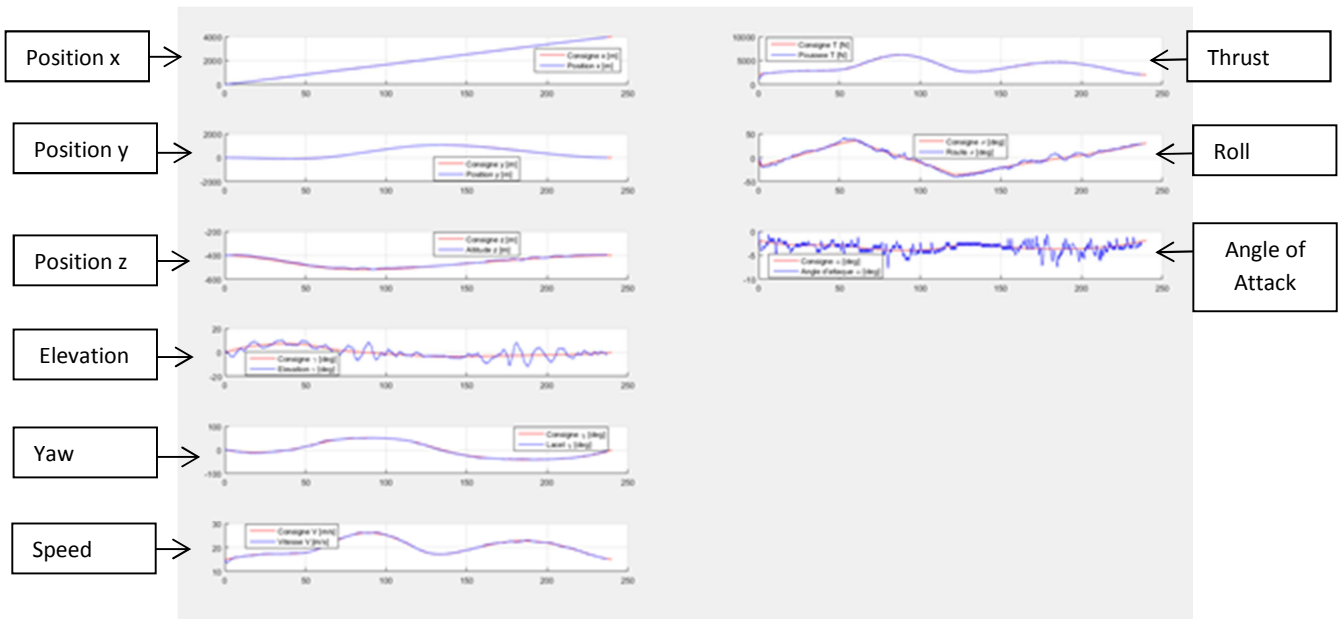


Figure 8 – Comparisons between command and response of system

Simulation results in satisfying matching between instructions and responses of system. Concerning incidence and elevation the differences are due to the significant inertia of the zeppelin model used in this simulation.

## 8. Conclusion

The developed solution shows satisfying results in the early verification and validation phase through simulations of some scenarios. Progress has to be performed on the variety of use cases as well as the implementation of complementary avoidance methods that could be more adapted to certain cases (speed, flight phase, amount of obstacles). Then the predictor will be further optimized in order to reach the safety level objective of the global function upon various criterions in addition to the ones already considered. For example, some criterion may be added to directly take into account the vehicle in-flight configuration and potential system failure cases.

Moreover, the detection part of the function needs to comply with the data inputs required by the avoidance to characterize the obstacles. Obstacles noise captured by the detection sensors may induce additional unnecessary multiple trajectory segments which will reduce function interest and generate additional energy related (I.e. fuel consumption) costs. Instead, optimization loops shall be carefully developed in order to issue proper trajectory management at least at a level that a human could have determined in a piloted mode. Safety and economical gains will be sufficiently significant when it will provide measurable added-value in representative scenarios compared to current trajectory management executed by a pilot.

The final developed system will then act as a path towards real autonomous “Detect & Avoid” and may generate concrete implementation opportunities on various supporting platforms such as drones or piloted aircraft, with, for instance, augmented reality head-up vision displaying safe trajectory superposed on visible environment.

## References:

- [1] Iwan Ulrich, Johann Borenstein, VFH\* : *Local Obstacle Avoidance with Look-Ahead Verification*, The University of Michigan, Advanced Technologies Laboratory, Ann Arbor, USA, 2000.
- [2] Simon Vanneste, Ben Bellekens, Maarten Weyn, 3DVFH+ : *Real-Time Three-Dimensional Obstacle Avoidance Using an Octomap*, CoSys-Lab, Faculty of Applied Engineering, Antwerpen, Belgique.
- [3] D. Vikenmark, *The Obstacle Restriction Method (ORM) for Reactive Obstacle Avoidance in Difficult Scenarios in Three Dimensional Workspaces*, Stockholm – Suede, 2006.
- [4] P. Fiorini <sup>(1)</sup>, Z. Shiller <sup>(2)</sup>, *Motion Planning in Dynamic Environments using Velocity Obstacle*, <sup>(1)</sup> California Institute of Technology, Pasadena – USA, <sup>(2)</sup> University of California, Los Angeles – USA.
- [5] Elie G Kahale, *Planification et Commande d'une Plate-Forme Aéroportée Stationnaire Autonome Dédiee à la Surveillance des Ouvrages d'Art*, Université d'Evry Val d'Essonne – France, 2014