A solution to detect and avoid conflicts for civil Remotely-Piloted Aircraft Systems into non-segregated airspaces

Giovanni Migliaccio, Giovanni Mengali and Roberto Galatolo

Department of Civil and Industrial Engineering, University of Pisa, Italy

Corresponding author:

Giovanni Mengali, Department of Civil and Industrial Engineering, University of Pisa, Via Girolamo Caruso, 8, 56122 Pisa, Italy.

Email: g.mengali@ing.unipi.it

Abstract

The capability to "detect and avoid" potential collisions is one of the main technical challenges restricting widespread operations of unmanned aircraft into non-segregated airspaces. In fact, to operate into prescribed environments, an unmanned aircraft needs an onboard technology to replace the capability of the human pilot to "see and avoid" collision hazards. Such a technology is a "sense and avoid" system. This paper focuses on the "avoid function" of such a system and proposes a suitable solution. The approach to the problem is to schematize a generic obstacle through a moving ellipsoid that represents the region of space the unmanned aircraft must not violate. The obtained solution enables situations of potential conflict to be detected and avoided through a set of possible actions such as speed changes in magnitude and/or direction. Thousands of test cases have been considered to validate this solution. Simulations show that the proposed algorithm is able to detect and avoid situations of potential conflict in the three-dimensional space and in real-time, even without the assistance of a human operator. As such, it can be considered as a fundamental step for the development of a prototype of "sense and avoid" system for promoting the integration of unmanned aircraft into non-segregated airspaces.

Keywords

Collision avoidance, traffic separation, unmanned aircraft

Introduction

One of the main technical challenges restricting operations of unmanned aircraft into the civil airspace beyond the visual line of sight is related to the capability to "detect and avoid" situations of potential conflict,[1-3] such as loss of separation or mid-air collision hazards. Fundamental requirements state in fact that an aircraft shall not be operated in such proximity to other aircraft as to create a collision hazard and vigilance shall always be maintained to "see and avoid" potential collisions.[4,5]

To operate into prescribed environments, both manned and unmanned aircraft have to comply with these requirements. However, while manned aircraft have the human pilot onboard that is also the ultimate responsible of taking actions to avoid collision hazards, unmanned aircraft do not always have the same possibility. For example, communication problems between an unmanned aircraft and the corresponding remote human pilot can jeopardize the flight safety. Therefore, at least in situations of this kind, a technology is required onboard the unmanned aircraft to replace the capability of the human pilot to "see and avoid" potential conflicts with an "equivalent level of safety". Such a technology is a so-called "sense and avoid" system.[6-10]

As far as the present research is concerned, the focus is on the development of a prototype of "avoid function", which is a function of a "sense and avoid" system. More precisely, a solution has been developed in terms of strategies and algorithms[11,12] that enable potential conflicts to be detected and resolved even without the assistance of a human operator.

Over the years several solutions have been proposed to automate the process of detecting and avoiding situations of potential collision. Reviews also exist that discuss, compare and classify different approaches for conflict detection and resolution, as well as evolving philosophies on collisions avoidance for unmanned aircraft.[13-15]

For example, there are methods based on the optimization of cost functions,[16-18] on potential fields,[19-20] on geometric considerations that define escape directions,[21-25] and many other approaches.[26-28]

As far as the optimization methods are concerned, they generally work by combining a model of the aircraft and a set of constraints in order to define a function that is used to find the optimal solution. For example, some methods make use of a node-based discretization of the space surrounding the aircraft and try to find a path from an initial to a final node using the minimum cost according to a specific metric. Among them, an example is represented by the A* algorithm.[18] A problem with these approaches is the balance between the resolution of the discretized space and the computational cost to achieve the solution. In fact, as the discretized space grows, the computational complexity increases too. On the other hand, if the airspace is broken into fewer larger cells, there might not be enough detail to find effective conflict-free paths.

Among the optimization methods, another powerful technique is the mixed integer linear programming (MILP). In that case, a system of linear constraints and an objective function must be passed to a solver software that is designed to find the solution.[16] Although a MILP approach can provide feasible solutions to collision avoidance problems, it is generally computationally expensive. Some strategies have also been proposed to reduce the time required to achieve feasible solutions,[17] but these approaches may also reduce the overall optimality of the MILP.

In general, the main drawback of the optimization approaches is the computational time, which is not predictable, and the convergence to a feasible solution, which is not always guaranteed in a finite time, or can be relatively slow. This makes such approaches tricky for real-time applications, especially when compared to other solutions.

The use of potential fields is another option.[14,19,20] In this case, either a moving aircraft or fixed obstacles are represented through charged particles and modified electrostatic equations are used to maneuver the reference aircraft away from potential hazards (repulsive points) while pointing to a goal (attraction point). Accordingly, a control action is continuously available to drive the aircraft. However, several problems may arise when these algorithms are used in practice, such as the possibility of running into local minima and the difficulty of obtaining feasible control actions. As a result, the corresponding maneuvers for conflict resolution are not always guaranteed to correspond to feasible results.[14]

Other diffused approaches are based on the use of geometric considerations to obtain analytical solutions to the problem of conflict resolution.[21-25] Some of them provide answers in the horizontal plane only,[21] others in the three-dimensional space too,[22] some of these make use of a cylinder to represent a hazard,[24] others are based on a safety sphere.[23] In general, geometric methods make use of kinematic information to define a condition for conflict detection, and find the corresponding actions (e.g. escape directions) for conflict resolution. The main idea behind the geometric approaches is to represent an intruder through a protected zone and calculate how its velocity vector should be modified to avoid a predicted conflict. They are relatively simple to develop, efficient and effective in proving collision avoidance solutions, and enable the constraints related to the feasibility of the control actions to be easily integrated within the solution. Fundamental advantages are also in the low computational effort and in the deterministic time they require to obtain feasible solutions even within a three-dimensional space. This feature make these methods especially suitable for real-time applications.

The analysis of the dedicated literature and the above considerations suggest the employment a geometric approach for the problem discussed here as a first attempt to obtain an analytical, deterministic and feasible solution for real time applications.

In particular, the proposed method may be thought of as a generalization of a solution available in the literature.[23] More precisely, the approach described in this paper is based on the concept of a generic moving ellipsoid that delimits a forbidden region of space that the reference unmanned aircraft cannot violate in order to avoid a situation of potential conflict with a generic obstacle or a moving intruder. The proposed solution enables the capability to detect and avoid situations of potential conflict to be implemented on-board and in real-time, using different possible actions that are well defined analytically, such as speed changes in intensity and/or direction, either with or without the assistance of a human operator.

The rest of the paper is organized as follows. The next section introduces the general problem considered in this work. Then, a solution is proposed to "detect and avoid" situations of potential conflict and the simulation environment within which this solution is implemented and validated is discussed. Finally, the results of a campaign of simulations are presented to demonstrate the effectiveness of the proposed solution to detect and avoid situations of potential conflict in real-time.

The "sense and avoid" problem

One of the main issues to be solved to integrate unmanned aircraft into non-segregated airspaces is related to the requirement for traffic separation and mid-air collision avoidance, also referred to as "sense and avoid" requirement.[6-10]

Generally speaking, the "sense and avoid" problem involves the development of solutions in terms of technologies, procedures and regulations, to make an unmanned aircraft able to implement the process of conflict detection and resolution even without the assistance of a human operator. More precisely, the term "sense" refers to the capability of acquiring information to detect potential hazards, while the term "avoid" refers to the capability of ensuring traffic separation and avoiding imminent collisions.

Several factors have to be considered to develop a "sense and avoid" solution, such as the environment in which the aircraft operates, the applicable rules and procedures,[4,5] the need of using different integrated sensor solutions to develop a "sense function" that meets different requirements,[32-33] the capability of an unmanned aircraft to perform specific maneuvers. In addition, for the problem of traffic separation, an important aspect is to identify how to share the responsibilities between the air traffic control (ATC) agent, the pilot in command (PIC) of a remotely piloted aircraft (RPA) and the "sense and avoid" system (SAAS) onboard. However, this is neither the most important objective nor it is mandatory when an imminent collision is predicted. In such a case the most important objective is the safety of the flight regardless of the flight rules and procedures.

Among these and other aspects of the problem, the present research focuses on the case in which the responsibility for separation and collision avoidance relies on the PIC or on the SAAS onboard the RPA. Moreover, for what concerns the main functions of a SAAS, this paper concentrates on the development of an "avoid function" only.

The "avoid function"

An "avoid function" shall be able to provide assistance to the PIC of a RPA to maintain the adequate minima of separation from the traffic. In addition, such a function shall also enable a RPA to avoid imminent collision hazards even without the assistance of a human operator (autonomous mode) at least in situations of emergency.

For what concerns the modes of operation of an "avoid function", both automatic and autonomous modes have to be considered. For example, during normal conditions of communication, the PIC of a RPA can use the available information to avoid a potential conflict either by configuring an automatic pilot system to execute a conflict avoidance maneuver, or by providing a clearance to the SAAS onboard the RPA to avoid a conflict through the solution the SAAS proposes. Instead, in case of communication problems between the PIC and the RPA, a SAAS onboard the RPA shall enable to operate the vehicle in a safe manner even without the assistance of the PIC.

Different constraints shall be considered to develop an "avoid function", such as the region of space a RPA should be able to monitor, the required minima of separation, the applicable flight rules and the procedures and the capability of a RPA to maneuver.

For example, a RPA should be able to monitor a region that extends ±110° in azimuth and ±15° in elevation.[8] Moreover, it should be able to maintain a safety distance from potential hazards, ensuring for example a miss distance from an intruder of 500 feet in the horizontal plane and 350 feet in the vertical plane.[6,7] Furthermore, a RPA should be able to execute maneuvers that comply with the rules of flight that apply in the specific situation (at least for problems of traffic separation). For example, to avoid a conflict with another aircraft, the RPA can be required to turn on the right or to maneuver in such a way to avoid passing in front of the other aircraft.[4] Finally, for what concerns the deviation of the RPA with respect to its pre-planned flight plan, a desirable objective is to execute maneuvers that minimize such a deviation while complying with the capability of the RPA of executing the required maneuvers.

The above constraints represent some of the main requirements to develop an "avoid function". A solution that is able to comply with the above constraints is now illustrated.

Conflict avoidance solution

To develop a conflict avoidance solution in terms of algorithms, a corresponding conflict avoidance strategy has been preliminary defined. Such a strategy takes into account the vision of ICAO about the process of "Conflict Management"[5] and is based on a set of key actions to be executed sequentially. These actions are the detection of a conflict, the formulation of a solution, the implementation of such a solution and the monitoring of the evolution of the situation. Moreover, the considered strategy enables the process of conflict detection and resolution to be schematized independently of the algorithms used to implement it (see Figure 1).

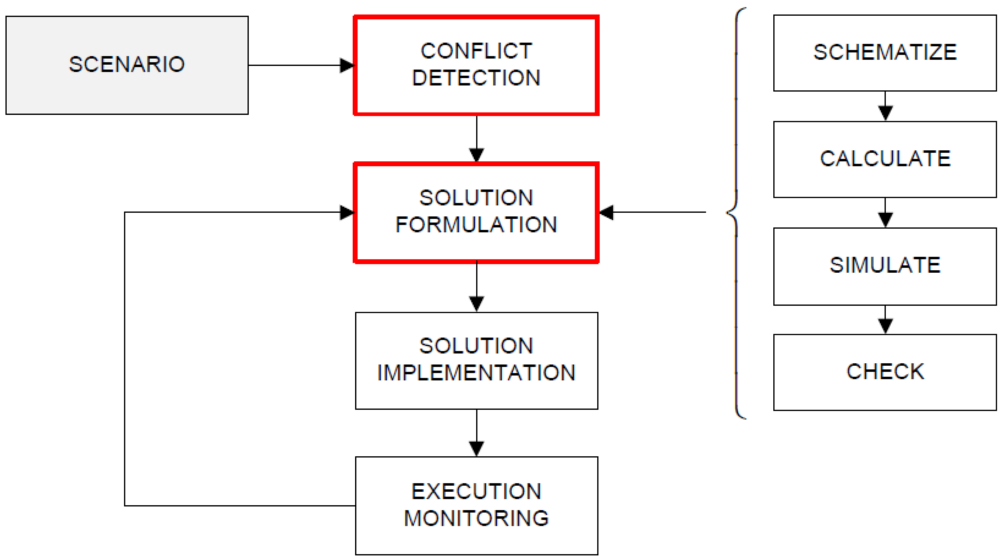

Figure 1. Conflict avoidance strategy

More precisely, the process evolves as follows. The first phase deals with the detection of the conflict. It involves sensors to acquire information from the environment and algorithms to identify the conflict. The second phase is the formulation of feasible solutions to avoid the predicted conflict. This can be done by sequential steps such as the schematization of the situation, the calculation of actions to avoid the detected conflict, the simulation of possible evolutions of the situation (with the use of the previous actions) and the verification of the effectiveness of the calculated solution to avoid the predicted conflict. Then, after a conflict has been detected and a solution has been formulated, it can be implemented. The final phase is the monitoring of the actual evolution of the situation. This is useful for dealing with unforeseen or unpredictable changes to the predicted trajectories of the aircraft, with the aim of providing additional feedbacks to the "avoid function" to effectively avoid the predicted conflict.

Conflict scenario schematization

To obtain a feasible, effective and practical solution in terms of algorithms, a reference schematization of a conflict scenario has been considered (see Figure 2).

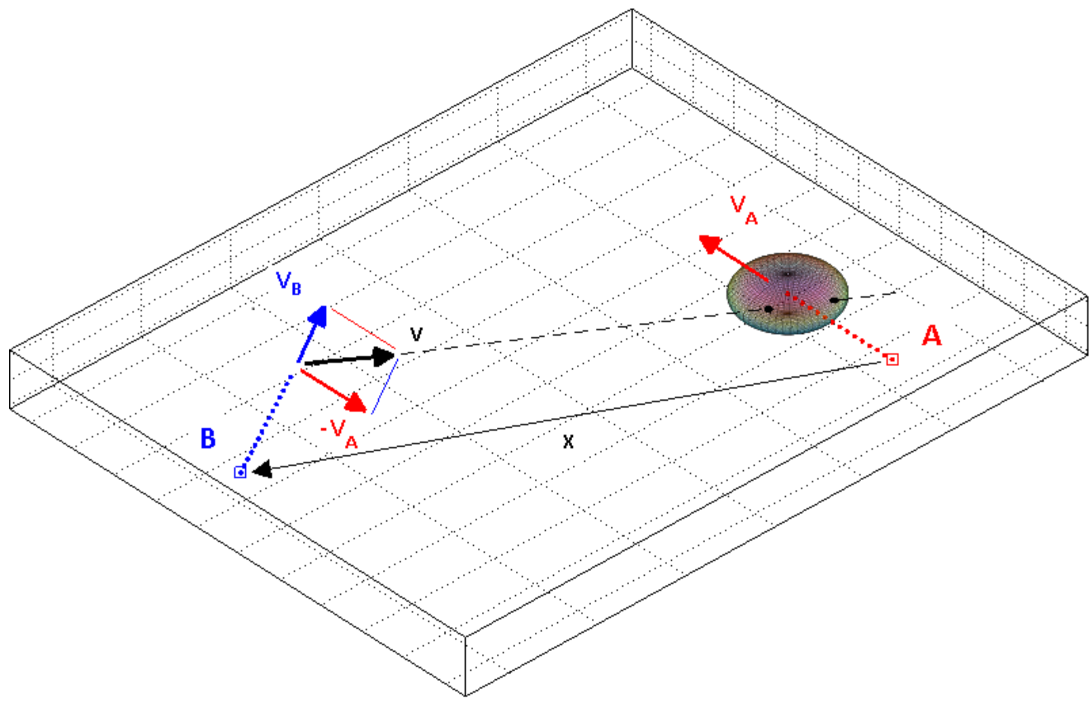

Figure 2. Conflict scenario schematization

The reference RPA is represented by the point-B, the intruder is represented by the point-A, and the ellipsoid centered at point-A represents a region of space around the intruder which the RPA must not violate.

According to the proposed schematization, a situation of conflict is predicted to occur if and only if the reference RPA is predicted to violate the ellipsoid of the intruder.

To avoid a predicted conflict, the RPA is assumed to execute a suitable maneuver. An algorithm has been developed to achieve this purpose even without the assistance of a human operator. Moreover, to obtain a solution to detect and resolve potential conflicts in the three-dimensional space and in real-time, the intruder is initially assumed to maintain its flight direction and speed. However, a continuous update of its actual position and velocity is executed to overcome the limitations related to such an assumption.

The conflict avoidance solution that has been obtained through the considered approach is introduced hereafter. Further details about the corresponding algorithms can be found within a previous work of the authors.[12]

Conflict detection and resolution algorithms

Over the years different approaches have been proposed to detect and resolve situations of potential conflict.[11-28] The present research is based on a geometric approach and uses the proposed schematization of the conflict scenario to define an analytical condition for conflict detection and an algorithm for conflict resolution.[11,12]

According to the considered approach, a situation of conflict is declared to occur if and only if the reference RPA is predicted to violate the ellipsoid of the intruder. To translate this statement into an analytical condition, consider for example the proposed schematization of the conflict scenario (see Figure 2), where the intruder is initially assumed to maintain its flight direction and speed. The current position and velocity of the RPA with respect to the intruder are **X** and **V**, respectively, and the semi-major axes of the safety ellipsoid are R_1, R_2, and R_3. Also, introduce a North-East-Down reference frame in the center of the safety ellipsoid (point A) and assume North, East and Down to be the first, second and third direction of such a reference frame. It may be easily verified that the reference RPA is outside the ellipsoid of the intruder if and only if the following inequality holds

$$\sum_{i=1}^{3} \frac{X_i^2}{R_i^2} > 1 \tag{1}$$

The left hand side of the inequality defines what is here referred to as "avoid distance" parameter. It is a non-dimensional parameter that can be used to define a useful metric to approach the problem of conflict detection and resolution.

According to the previous schematization, it can be shown that if the reference RPA is outside the ellipsoid of the intruder, a potential conflict is predicted to occur if and only if the following condition holds

$$\sum_{i=1}^{3} \frac{X_i V_i}{R_i^2} + \sqrt{\sum_{i=1}^{3} \frac{V_i^2}{R_i^2} \left(\sum_{i=1}^{3} \frac{X_i^2}{R_i^2} - 1 \right)} < 0 \tag{2}$$

To avoid a situation of predicted conflict, the reference RPA is assumed to execute a conflict avoidance maneuver according to a specifically developed algorithm, which is now summarized.[12]

The quadratic form associated to the intruder's ellipsoid is modeled through a positive definite matrix $\boldsymbol{\sigma} \in [R^3 x R^3]$ whose eigenvalues are

$$\mathrm{eig}_i(\boldsymbol{\sigma}) = 1 / R_i^2 \quad i = 1,2,3 \tag{3}$$

Let S be a vector space associated to the physical space, where $\mathbf{p_1}$ and $\mathbf{p_2}$ are two homogeneous vectors, but $\mathbf{p}$ and $\mathbf{q}$ need not to be homogeneous. The inner-product, norm and distance are defined in S as:

$$s(\cdot,\cdot) : SxS \to R \,|\quad s(\mathbf{p},\mathbf{q}) = \mathbf{p}^T \boldsymbol{\sigma} \mathbf{q} \tag{4}$$

$$n(\cdot) : S \to R^+ \,|\quad n(\mathbf{p}) = \sqrt{s(\mathbf{p},\mathbf{p})} \tag{5}$$

$$d(\cdot,\cdot) : SxS \to R \,|\quad d(\mathbf{p_1},\mathbf{p_2}) = n(\mathbf{p_2} - \mathbf{p_1}) \tag{6}$$

If $\mathbf{r}$ is a generic vector centered at A, the conflict region can be mathematically characterized as

$$d(\mathbf{r},\mathbf{0}) < 1 \tag{7}$$

It can be shown that if $\mathbf{x}$ and $\mathbf{v}$ represent the generic relative position and velocity vectors of B with respect to A at a time instant τ, a general condition for collision avoidance is

$$D(\mathbf{x},\mathbf{v}) \leq 0 \tag{8}$$

where

$$\mathrm{D}(\mathbf{x},\mathbf{v}) = \mathrm{s}(\mathbf{x},\mathbf{v})\mathrm{s}(\mathbf{v},\mathbf{x}) - \mathrm{s}(\mathbf{v},\mathbf{v})[\mathrm{s}(\mathbf{x},\mathbf{x}) - 1] \tag{9}$$

The condition (8) can be used in a constructive way to calculate the velocity variations to avoid a predicted collision between aircraft B and intruder A. In particular, we concentrate on a set of trajectories that are tangent in a time-space domain to the safety ellipsoid of the intruder A.

To achieve this, consider an (impulsive) velocity change $\Delta\mathbf{v}$ such that

$$\mathbf{v} = v_0\,\mathbf{u}_\mathrm{r} + \Delta\mathbf{v} \mid \quad \mathrm{D}(\mathbf{x},\mathbf{v}) = 0 \tag{10}$$

where $v_0\mathbf{u_r}$ is the relative velocity vector before the velocity change $\Delta\mathbf{v}$ is applied (v_0 is the intensity of such a velocity and $\mathbf{u_r}$ is a unit vector that represent its direction).

Two special solutions of equation (10) are illustrated. The first one (when applicable) involves a velocity change parallel to the initial velocity vector of aircraft B (**case 1**). The second solution involves a velocity change in both modulus and direction (**case 2**).

For what concerns the first solution (**case 1**), consider a speed increment in the form

$$\Delta\mathbf{v} = k\,v_0\mathbf{e} \mid \quad \mathbf{e} = \pm\mathbf{u}_\mathrm{B} \tag{11}$$

where k is a real number, and $\mathbf{u_B}$ is a unit vector with the same direction of the initial velocity of aircraft B. Solving equation (11) with respect to this $\Delta\mathbf{v}$ yields

$$\Delta\mathbf{v} = \left[\frac{\mathbf{e}^\mathrm{T}\mathbf{M}\mathbf{u}_\mathrm{r}}{-\mathbf{e}^\mathrm{T}\mathbf{M}\mathbf{e}} + \sqrt{\left(\frac{\mathbf{e}^\mathrm{T}\mathbf{M}\mathbf{u}_\mathrm{r}}{-\mathbf{e}^\mathrm{T}\mathbf{M}\mathbf{e}}\right)^2 + \frac{\mathbf{u}_\mathrm{r}^\mathrm{T}\mathbf{M}\mathbf{u}_\mathrm{r}}{-\mathbf{e}^\mathrm{T}\mathbf{M}\mathbf{e}}}\right] v_0\mathbf{e} \tag{12}$$

where

$$\mathrm{M}(\mathbf{x}) = \boldsymbol{\sigma}\mathbf{x}\mathbf{x}^{\mathrm{T}}\boldsymbol{\sigma} - (\mathbf{x}^{\mathrm{T}}\boldsymbol{\sigma}\mathbf{x} - 1)\boldsymbol{\sigma} \tag{13}$$

Next (**case 2**), consider a speed increment in the form

$$\Delta\mathbf{v} = k\, v_0\, \mathbf{w}(\delta,\theta) \tag{14}$$

in which **w** is a unit vector defined as

$$\mathbf{w}(\delta,\theta) = \mathrm{Sin}\,\theta\,\mathbf{u}_{\mathrm{r}} + \mathrm{Cos}\,\theta \cdot (\mathrm{Cos}\,\delta\,\mathbf{u}_{\mathrm{s}} + \mathrm{Sin}\,\delta\,\mathbf{u}_{\mathrm{r}} \times \mathbf{u}_{\mathrm{s}}) \tag{15}$$

where $\delta \in [0, 2\pi]$ and $\theta \in [-\pi/2, \pi/2]$ are generic angles, and the unit vector $\mathbf{u}_{\mathrm{s}}$ is orthogonal to the unit vector $\mathbf{u}_{\mathrm{r}}$. Algebraically solving equation (10) with respect to the velocity variation of equation (14) yields

$$\Delta\mathbf{v} = \frac{\mathrm{Cos}(\theta_1)}{\mathrm{Sin}(\theta_1 - \theta_2)} v_0 \mathbf{w}(\delta,\theta_1) \tag{16}$$

in which θ_1 and θ_2 have to be calculated according to the following algorithm

$$\rho_1 = \frac{\mathbf{w}(\delta,0)\mathbf{M}\mathbf{w}(\delta,\pi/2)}{-\mathbf{w}(\delta,0)\mathbf{M}\mathbf{w}(\delta,0)} \tag{17}$$

$$\rho_2 = \frac{\mathbf{w}(\delta,\pi/2)\mathbf{M}\mathbf{w}(\delta,\pi/2)}{-\mathbf{w}(\delta,0)\mathbf{M}\mathbf{w}(\delta,0)} \tag{18}$$

$$\theta_1 = \mathrm{Cot}^{-1}\left[\rho_1 + \sqrt{\rho_1^2 + \rho_2}\right] \tag{19}$$

$$\rho = v_0 / V_B \tag{20}$$

$$\rho_A = V_A / V_B \tag{21}$$

$$\rho_B = V_{B*} / V_B \tag{22}$$

$$\rho_3 = \rho_A \mathbf{u}_{\mathrm{A}}^{\mathrm{T}} \mathbf{w}(\delta,\theta_1) \tag{23}$$

$$\theta_2 = \mathrm{Tan}^{-1}\left(\mathrm{Tan}\,\theta_1 - \frac{\rho / \mathrm{Cos}\,\theta_1}{\rho_3 + \sqrt{\rho_3^2 + \rho_B^2 - \rho_A^2}}\right) \tag{24}$$

and where V_B and V_{B*} are, respectively, the speeds of the aircraft B before and after the conflict avoidance maneuver, V_A is the speed of the intruder A, and $\mathbf{u_A}$ is a unit vector with the direction of flight of the intruder A. Note that an appropriate value for the final velocity V_{B*} has to be chosen to calculate the parameter of equation (22). For example, if a maneuver at a constant speed (intensity) can avoid a predicted conflict, a useful choice can be $V_{B*} = V_B$, otherwise a change in the speed intensity is also required.

The proposed algorithms provide the changes in the speed vector of the reference RPA that can be implemented by an automatic pilot system to avoid a potential conflict even without the assistance of a human operator. Further details about the proposed solution can be found elsewhere.[11,12]

These algorithms have been integrated within a simulation environment that is now illustrated.

Simulation environment

A campaign of simulations has been conducted to validate the proposed solution. They involve both human-in-the-loop (HIL) and fast-time (FT) simulations. To achieve this, a simulation environment has been developed.

Simulation models and user interfaces

The simulation environment is based on simulation models that have been developed, implemented and integrated. Within this environment, an aircraft is modelled as a rigid body with six degrees of freedom. For the purpose of this paper, the environment contains a reference RPA and an intruder (see Figure 3).

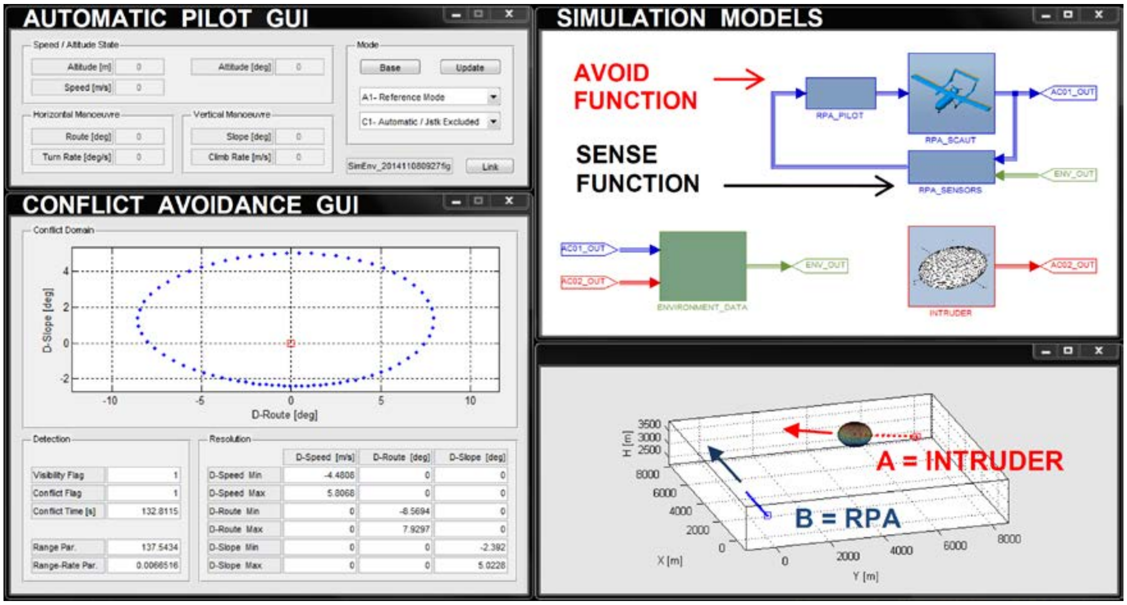

Figure 3. Simulation Environment with RPA and INTRUDER

The intruder is modelled with a point mass, corresponding to the center of an ellipsoid the reference RPA must not violate. The RPA is instead represented through a model that simulates its dynamics as a rigid body with six degrees of freedom.

Additional models have also been added to the simulation environment. For example, a model of the sensors of the RPA simulates the process of acquiring information about potential obstacles in the scenario. A model of the automatic pilot system of the RPA and a corresponding graphical user interface (i.e. the Automatic Pilot GUI) enable automatic maneuvers to be performed. A model of the "avoid function" of the RPA is used to detect and avoid potential conflicts both with and without the assistance of a human operator. A graphical user interface (i.e. the Conflict Avoidance GUI) enables the PIC to utilize information from the "avoid function" to detect and avoid a potential conflict, while autonomous operations are enabled through a direct connection between the "avoid function" and the automatic pilot system of the RPA.

Conflict avoidance tool

The Automatic Pilot GUI, the Conflict Avoidance GUI and the "avoid function" define what is here referred to as Conflict Avoidance Tool. The integration of these components is a fundamental step to demonstrate how the PIC of a RPA can utilize the information from the proposed solution to detect and avoid situations of potential conflict in real-time. More precisely, the Automatic Pilot GUI is a simplified representation of a possible interface between the PIC and the automatic pilot system of the RPA (it allows an user to utilize the automatic pilot system to perform automatic maneuvers). Instead, the Conflict Avoidance GUI is a simplified representation of a possible interface between the PIC and the "avoid function" onboard the RPA (it allows an user to acquire information about a potential conflict and indications to avoid it, such as changes in speed module, route angle and/or slope angle).

In general, the developed "avoid function" is able to provide generic (mixed) actions of conflict resolution. Some of them are especially suitable for an implementation within a standard automatic pilot system and are suggested to the remote human operator by means of the proposed Conflict Avoidance GUI. On the contrary, the complete set of all possible avoidance actions could be better managed by an automated system whose function is to decide which action should be taken to prevent a conflict occurrence. Of course, other possibilities could be considered for connecting the developed "avoid function" and the human operator, at the expense of increasing the complexity of the conflict avoidance system in terms of human-machine interface.

For what concerns the proposed Conflict Avoidance GUI, it is constituted by some main panels (see Figure 4): namely the detection panel, the resolution panel and the conflict domain panel. The detection panel provides information about potential conflicts. For example, the "conflict flag" indicates the existence of a potential conflict (1 conflict, 0 no conflict), while the "conflict time" indicates the time before this conflict is predicted to occur. The resolution panel contains a table with indications on how to resolve a situation of predicted conflict. Each row of the table contains a solution in terms of minimum and maximum changes in speed module, route angle, and/or slope angle. More precisely, the first and second row provide the minimum and maximum changes in speed module only (if any). The third and fourth row provide the minimum and maximum changes of route angle at constant speed. The fifth and sixth row give the minimum and maximum changes of slope angle at constant speed. Moreover, if a path deviation at constant speed is not possible, the table also suggests the changes in speed intensity to be applied as well as the change of the considered path angle (route or slope).

Finally, within the conflict domain panel, the points outside the conflict domain boundary represent the deviations in both the flight path angles (route and slope) for implementing a conflict avoidance maneuver. More precisely, the more the deviations are outside the conflict domain, the larger the miss distance between the RPA and the intruder will be after the conflict avoidance maneuver. Instead, points on the boundary of the conflict domain correspond to maneuvers that will lead the RPA to avoid a situation of conflict by remaining tangent to the intruder ellipsoid (these are points of minimum deviation in terms of flight path angles at a given speed).

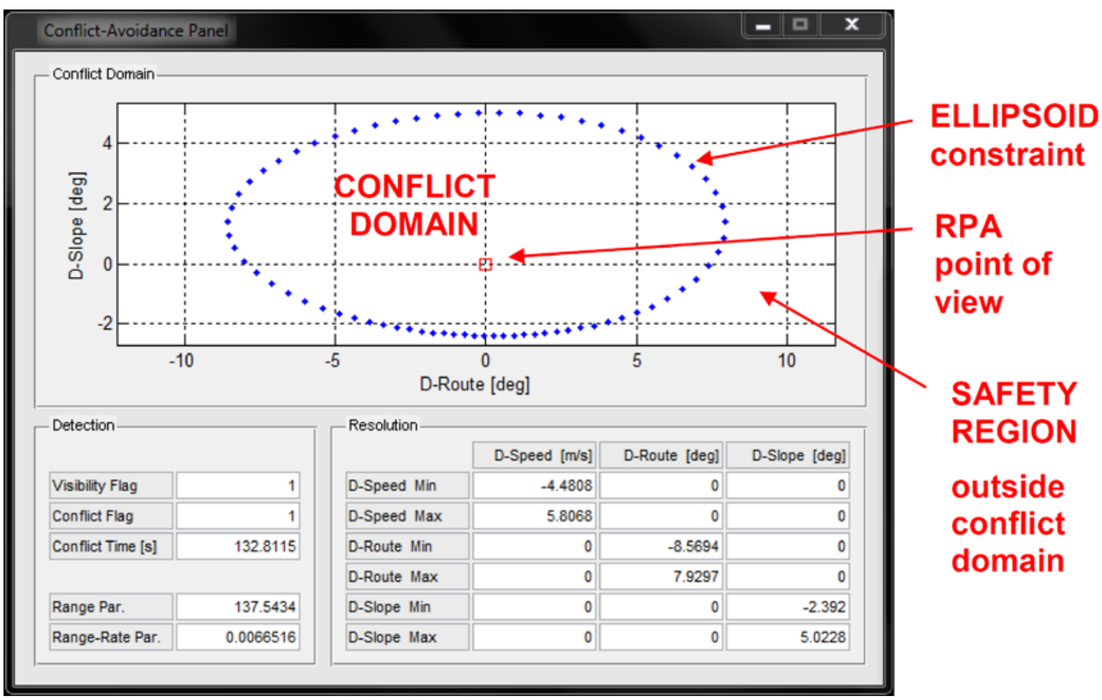

Figure 4. Conflict Avoidance GUI: interface with the "Avoid Function"

Simulation results

The validation of the proposed solution has been carried out via numerical simulations based on several conflict situations. The test cases involve human-in-the-loop simulations and fast-time simulations. HIL simulations demonstrate that the proposed solution can provide assistance to the PIC of a RPA during situations of potential conflict. Instead, FT simulations provide a wider assessment of the solution by means of a larger set of test cases in terms of initial positions and velocities of the intruder with respect to the RPA.

Human-In-The-Loop Simulation

The proposed test case involves the reference RPA in cruise at an altitude of 3000 m and speed of 45 m/s, while the corresponding intruder is in cruise at an altitude of 3000 m and speed of 50 m/s. The relative positions and directions of flight of RPA and intruder are illustrated in Figure 3. The intruder is also the center of a safety region modelled as an ellipsoid the RPA must not violate. The values of the semi-major axes of this ellipsoid are assumed to be R_1=R_2=600 m in the horizontal plane and R_3=420 m in the vertical plane. These values correspond to the application of a safety factor of about four with respect to the values of miss distances of 500 ft and 350 ft, respectively.

The proposed situation corresponds to a case in which a potential conflict is predicted to occur within 180 seconds. When such a conflict is detected, information about it is provided to the human operator through the detection panel and the resolution panel of the Conflict Avoidance GUI. The human operator may use the proposed solutions to implement a maneuver to avoid the detected conflict through the automatic pilot system of the RPA. In the considered test case, the selected maneuver is a positive change of the route angle of the RPA and it is implemented about 70 seconds after the conflict detection (see Figure 5).

As a consequence of the conflict avoidance maneuver, the RPA begins to turn on its right at a constant speed to achieve the desired conflict free route (see Figure 5). A few seconds after the minimum miss distance between the RPA and the intruder is reached, another maneuver is implemented by the human operator to drive the RPA on its pre-planned flight path (see Figure 5).

The results of several HIL simulations (similar to the one illustrated here) demonstrate that the PIC of a RPA could actually and effectively use the proposed conflict avoidance solution (by means of suitable graphical user interfaces) to detect and avoid situations of potential conflict within a tree-dimensional space in real-time.

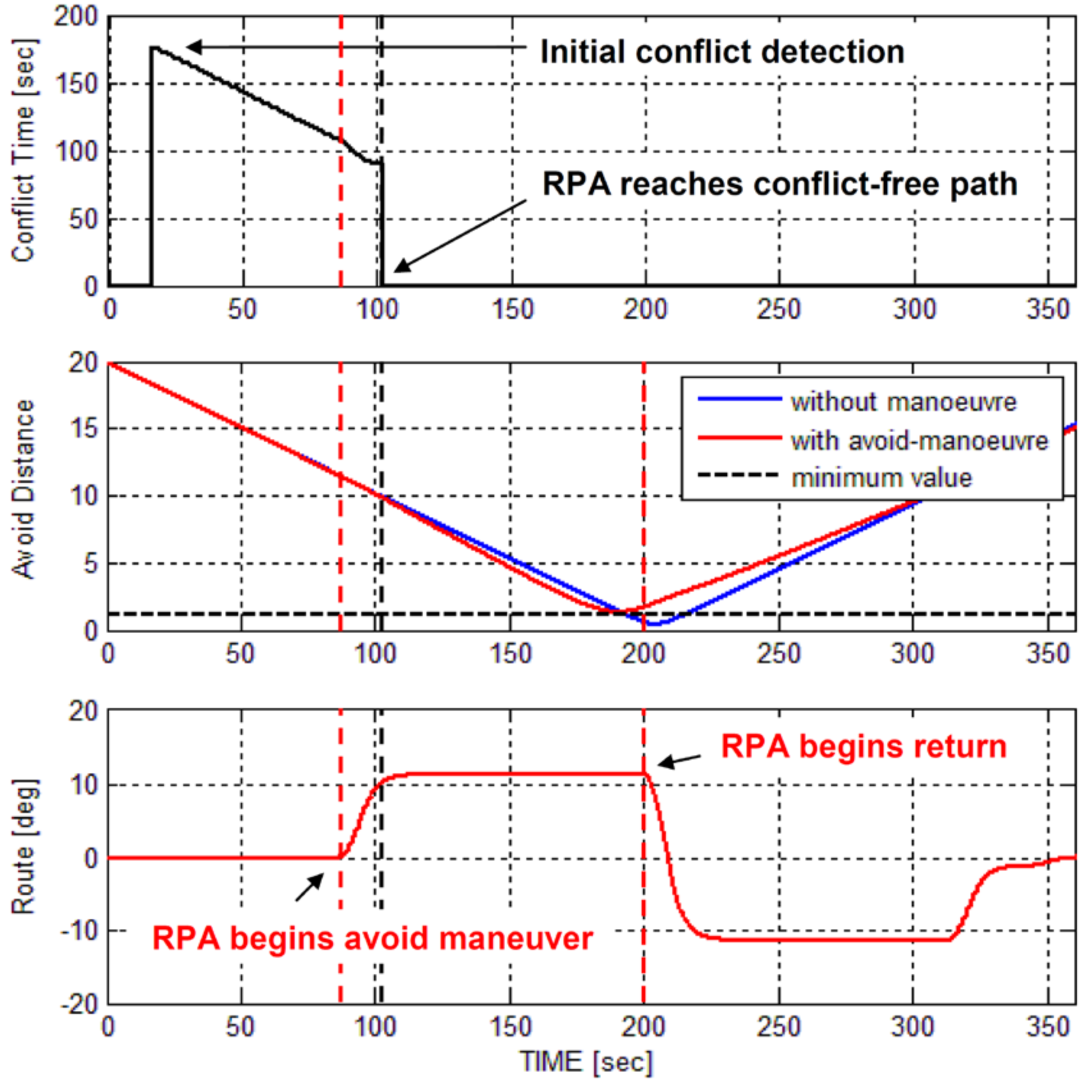

Figure 5. Simulation Time-History: Conflict Time, Avoid Distance, Route Angle

Fast-Time Simulation

To provide a wider assessment of the proposed solution, a more comprehensive set of test cases has been considered and fast-time simulations have been performed with the SAAS onboard the RPA in autonomous mode.

Different conflict geometries have been considered (see Figure 6). More precisely, the RPA always starts from the same condition, that is, a condition of cruise at an altitude of 3000 m and a speed of 45 m/s. The intruder is assumed to be in a flight phase in which it is following a given direction (in the horizontal plane or in the vertical plane) at a given speed. Then, the whole set of conflict geometries has been obtained by varying the initial positions and velocities of the intruder with respect to the reference RPA, as is shown in Figure 6. The initial distance between the intruder and the RPA is always about 5 nautical miles, while their relative positions (both in the horizontal plane and in the vertical plane) vary as illustrated in Figure 6. The velocity vector of the intruder also varies, both in module (between 40 m/s and 80 m/s) and in direction (between -1 and +1 degrees of orientation in the vertical plane and with different orientations in the horizontal plane), see Figure 6.

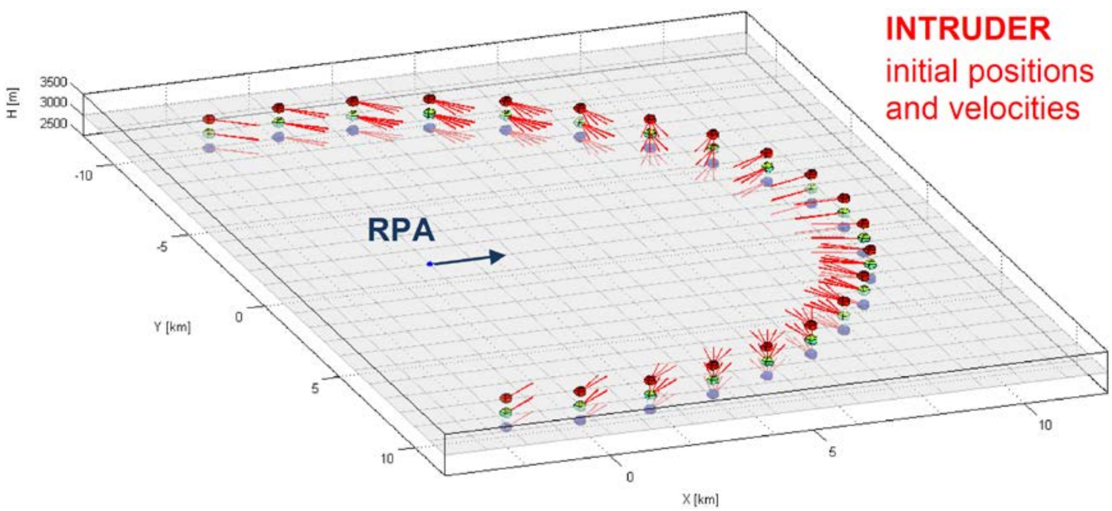

Figure 6. Fast-Time simulation cases: conflict geometries

The number of simulations is very high in this case, and an illustration of all of the corresponding time histories would not be feasible. To overcome this problem, the results have been organized into maps of parameters, where the parameters of interest are drawn against the initial values of the horizontal angular position of the intruder with respect to the RPA (see Figures 7 to 10). Note that the central zone into these maps does not contain any point because the corresponding angular positions are external to the field of view of the RPA and they have not been considered in these simulations.

Among all the parameters of interest, some reference maps are illustrated here. These are the two maps of the minimum avoid distance parameter (see Figures 7 and 8) and the two maps of the minimum relative distance between the reference RPA and the intruder (see Figures 9 and 10).

The maps of the avoid distance parameter allows one to quickly verify whether the RPA remained outside the safety ellipsoid during all the considered simulations.

For example, Figure 7 illustrates what happens in terms of minimum avoid distance when the reference RPA does not execute a maneuver to avoid the conflict (i.e. the conflict takes place). Instead, Figure 8 illustrates what happens in terms of minimum avoid distance if the reference RPA executes a maneuver according to the proposed avoidance solution. In this latter case the minimum avoid distance parameter (at the closest point of approach) is greater than one during all the considered simulations.

This demonstrates the effectiveness of the proposed conflict avoidance solution to detect and avoid situations of potential conflict even without the assistance of a human operator. As a matter of fact the RPA remains outside the ellipsoid of the intruder during all the simulations where the avoidance solution is used (see Figures 8 and 10).

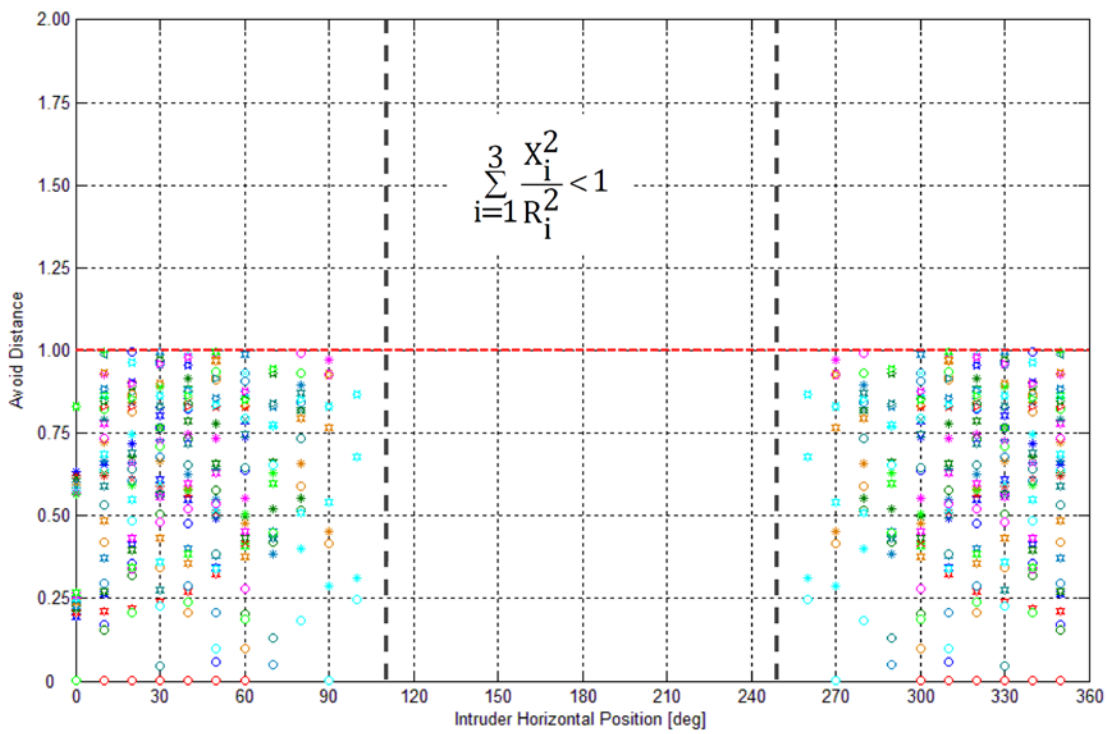

Figure 7. Simulation Map: minimum avoid distances without maneuvers

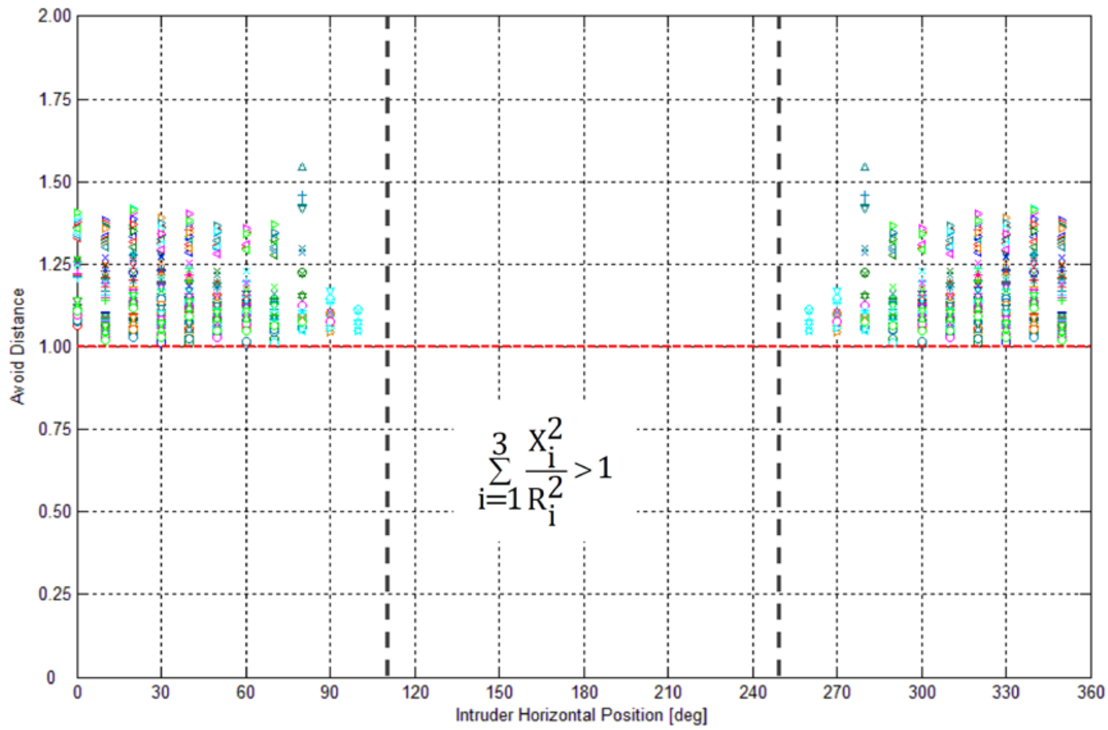

Figure 8. Simulation Map: minimum avoid distances with avoid maneuvers

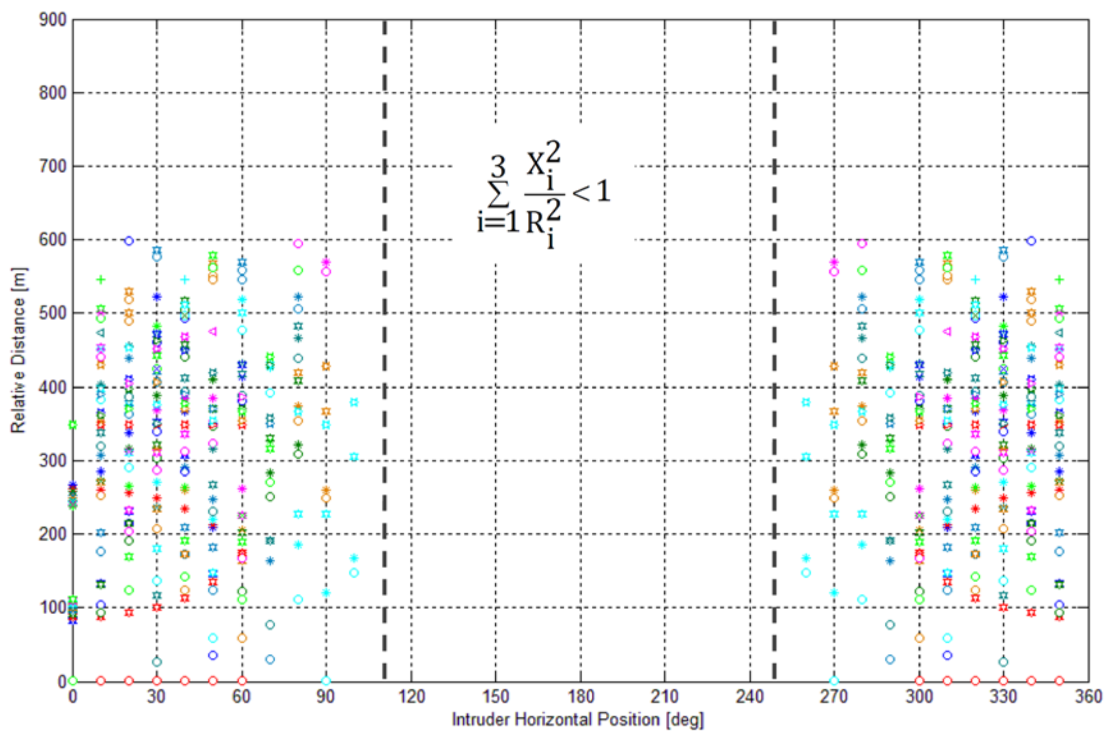

Figure 9. Simulation Map: minimum relative distances without maneuvers

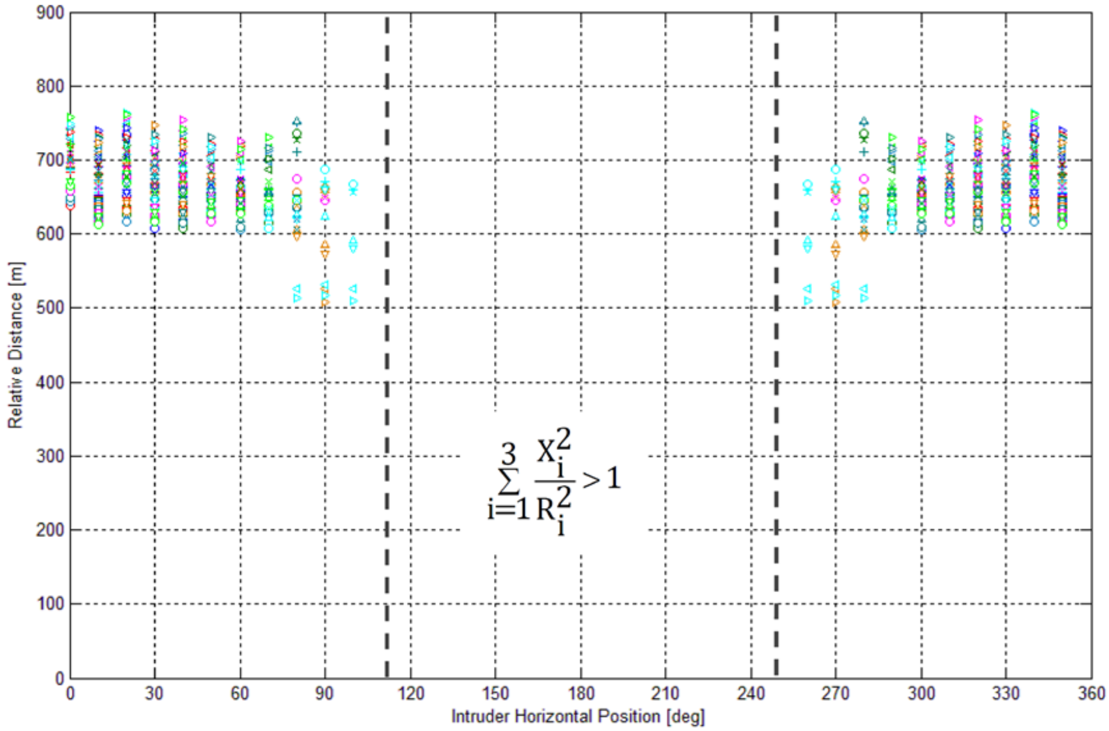

Figure 10. Simulation Map: minimum relative distances with avoid maneuvers

Conclusions

To operate into the civil non-segregated airspace in a way similar to a manned aircraft, a RPA needs an onboard system which guarantees the capability of detecting and avoiding potential conflicts to comply with the "sense and avoid" requirement.

Within the problem of developing a "sense and avoid" system, this paper has focused on the "avoid function" and proposed a solution to give assistance to the PIC of a RPA (to guarantee that the traffic separation be maintained) and to operate the RPA in a safe manner even without the assistance of a human operator at least in situations of emergency (to guarantee that imminent collision hazards be avoided).

To achieve this purpose, a strategy has been defined to detect and resolve situations of potential conflict and suitable algorithms have been developed to implement such a strategy. In addition, to demonstrate the effectiveness of the proposed solution, a set of simulation models have been developed, including the model of the reference RPA, the model of an intruder and some graphical user interfaces that enable the user to perform human-in-the-loop simulations. These models have been collected into an integrated tool, thus defining a simulation environment useful for validating the proposed solution by means of human-in-the-loop and fast-time simulations. The proposed solution was shown to be able to detect and resolve situations of potential conflicts in a three-dimensional space and in real-time, enabling the reference RPA to operate in a safe manner even without the assistance of a human operator.

To obtain a more advanced prototype of "avoid function" further developments are still necessary. For example, the proposed model of "avoid function" should be integrated, in the simulation environment, with a representative model of a "sense function" which takes into account the uncertainties related to the measurements of position and velocities of aircraft, intruders and generic obstacles. Other considerations involve the possibility that conflicting aircraft could be equipped with the same collision avoidance system. In that case, a suitable logic should also be agreed to ensure compatible maneuvers for those aircraft.

The considered problem is complex both from a regulatory and a technical point of view and an iterative approach is needed to achieve a mature solution. The solution proposed in this paper can be thought of as a useful starting point to develop a prototype of "sense and avoid" system for promoting the integration of unmanned aircraft into not-segregated airspaces.

References

1. ERSG. Roadmap for the integration of civil remotely-piloted aircraft systems into the European aviation system. Final Report from the European RPAS Steering Group, June 2013.

2. FAA. Integration of civil unmanned aircraft systems (UAS) in the national airspace system (NAS) roadmap. 1st Ed., 2013.

3. ICAO. Unmanned aircraft systems (UAS). Cir. 328, 2011.

4. ICAO. Annex 2 to the convention on civil aviation - rules of the air. 10th Ed., 1990.

5. ICAO. Global air traffic management operational concept. Doc. 9854, 1st Ed., 2005.

6. EUROCONTROL. Specifications for the use of military unmanned aerial vehicle as operational traffic outside segregated airspaces. July 2007.

7. NATO. Sense and avoid requirements for unmanned aerial vehicle systems operating in non-segregated airspace. NATO Unclassified, April 2008.

8. ASTM F2411-07. Standard specification for design and performance of an airborne sense and avoid system. ASTM International, West Conshohocken, PA, 2007.

9. FAA. Sense and avoid (SAA) for unmanned aircraft system (UAS). Final Report of the FAA SAA sponsored workshop, October 2009.

10. Prats X., Delgado L., Ramírez J., et al. Requirements, issues, and challenges for sense and avoid in unmanned aircraft systems. *Journal of Aircraft*, Vol. 49, No. 3, June 2012.

11. Migliaccio G., Mengali G. and Galatolo R. Traffic separation and collision avoidance strategies to promote civil applications of UAVs. In: *proceedings of the 3rd international conference on Application And Theory of Automation in Command And Control*, Naples, Italy, 28-31 May, ACM, New York, USA, pp. 12-32, 2013.

12. Migliaccio G., Mengali G. and Galatolo R. Conflict detection and resolution algorithms for UAVs collision avoidance. *The Aeronautical Journal*, Vol.118, N.1205, pp. 828-842, July 2014.

13. Kuchar J.K. and Yang L.C. A review of conflict detection and resolution modelling methods. *IEEE Transaction on Intelligent Transportation Systems*, Vol. I, No. 4, pp. 179-189, 2000.

14. Mujumdar A. and Padhi R. Evolving philosophies on autonomous obstacle/collision avoidance of unmanned aerial vehicles. *Journal of Aerospace Computing, Information, and Communication*, Vol. 8, No 2, pp. 17-41, 2011.

15. Holt J., Biaz S., and Aji C.A., Comparison of unmanned aerial system collision avoidance algorithms in a simulated environment. *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 3, pp. 881-883, May–June 2013.

16. Schouwenaars T., Moor B., Feron E., et al. Mixed integer programming for multi-vehicle path planning. In: *Proceedings of the European Control Conference*, pp. 2603-2608, 4-7 September 2001.

17. Schouwenaars T., Ferony E. and How J. Safe receding horizon path planning for autonomous vehicles. In: *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing,* October 2002.

18. Ruz J. J., Arevalo O., Pajares G., et al. Decision making among alternative routes for UAVs in dynamic environments, *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 326-331, September 2007.

19. Paul T., Krogstad T.R. and Gravdahl J.T. UAV formation flight using 3D potential field. In: *Proceedings of the 16th Mediterranean Conference on Control and Automation*, Ajaccio, France, 25-27 June 2008.

20. Sigurd K. and How J. UAV trajectory design using total field collision avoidance. *AIAA Guidance, Navigation, and Control Conference and Exhibit,* Austin, Texas, 11-14 August 2003.

21. Bilimoria K.D. A geometric optimization approach to aircraft conflict resolution. In: *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, Colorado, 14-17 August 2000.

22. Carbone C., Ciniglio U., Corraro F., et al. A novel 3D geometric algorithm for aircraft autonomous collision avoidance. In: *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 1580-1585, December 2006.

23. Luongo S., Carbone C., Corraro F., et al. An optimal 3D analytical solution for collision avoidance between aircraft. *2009 IEEE Aerospace Conference,* Big Sky, MT, March 2009.

24. Corraro F., Ciniglio U., Di Vito V., et. al. A novel 3D analytical algorithm for autonomous collision avoidance considering cylindrical safety bubble. *2010 IEEE Aerospace Conference*, Big Sky, MT, March 2010.

25. Chakravarthy A. and Ghose D. Obstacle avoidance in a dynamic environment: a collision cone approach. *IEEE Transaction on Systems, Man, and Cybernetics, Part A: Systems and Humans*, Vol. 28, No. 5, September 1998.

26. Temizery S., Kochenderfer M.J., Kaelbling L.P., et al. Collision avoidance for unmanned aircraft using Markov decision processes. *AIAA Guidance, Navigation, and Control Conference,* Toronto, Canada, 2-5 August 2010.

27. Izadi H.A., Gordon B.W. and Zhang Y. Rule-Based Cooperative collision avoidance using decentralized model predictive control. *AIAA Infotech@Aerospace Conference*, St. Louis, Missouri, 29-31 March 2011.
28. Kochenderfer M.J., Chryssanthacopoulos J.P. and Weibel R.E. A new approach for designing safer collision avoidance systems. *9th USA/Europe Air Traffic Management Research and Development Seminar*, Berlin, Germany, 14-17 June 2011.
29. Fasano G., Accardo D., Moccia M., et al. Multi-sensor-based fully autonomous non-cooperative collision avoidance system for unmanned aerial vehicles. *Journal of Aerospace Computing, Information and Communication*, Vol. 5, No. 10, pp. 338-360, 2008.
30. Rousseau M., Ratton L. and Fournet T. Multiple sensor tracking in a sense and avoid context. In: *27th International Congress of the Aeronautical Sciences*, Nice, France, 19-24 September 2010.