27

UAV Trajectory Planning for Static and Dynamic Environments

José J. Ruz[1], Orlando Arévalo[1], Gonzalo Pajares[2] and Jesús M. de la Cruz[1]
[1]*Dept. of Computer Architecture and Automatic Control, Complutense University of Madrid*
[2]*Dept. of Software Engineering and Artificial Intelligence, Complutense University of Madrid*
Spain

1. Introduction

An unmanned aerial vehicle (UAV) is a robotic aircraft that can fly with either a remote input from a ground-based operator, or autonomously without human intervention based on pre-programmed flight plans (How et al., 2004). UAVs offer advantages over conventional manned vehicles in many applications because they can be used in situations otherwise too dangerous for manned vehicles and without being weighed down by the systems required by a pilot. UAVs are currently receiving much attention in research because they can be used in a wide variety of fields, both civil and military, such as reconnaissance, geophysical survey, environmental and meteorological monitoring, aerial photography, and search-and-rescue tasks. Most of these missions are usually carried out in threatened environments, and then it is very important to fly along a route which keeps the UAV away from known threats. Detection radars are one of the main threats for an UAV, but there are others that should also be avoided, such as fires, electric storms, radio shadowing zones, no flight zones, and so on. One of the main goals in many UAV's projects has been to establish the route that maximizes the likelihood of successful mission completion taking into account all known information about technological constraints, obstacles and threat zones on a static environment (Richards & How, 2002). Some papers that investigate path planning for UAVs presume that the location of the threats and their presence are deterministically known at planning-time, and interpret a path which avoids possible threat regions as an optimal path (Borto, 2000). However more recent projects are examining the possibilities of UAVs as realistic autonomous agents working on dynamic environments where threat zones called pop-up are present (Zengin & Dogan, 2004). The true presence of these types of zones is only known at flying-time, but the location and knowledge about the probability of appearance can be known at planning-time.

In this chapter we will present an approach to trajectory optimization for UAV in presence of obstacles, waypoints, and risk zones. The approach has been implemented on SPASAS (*System for Planning And Simulation of Aerial Strategy*), an integrated system for definition of flight scenarios, flight planning, simulation and graphic representation of the results developed at Complutense University of Madrid. The system uses two alternative methods for trajectory generation: mixed integer linear programming (MILP) and a modification of the A* algorithm, depending on the characteristics of the scenario between two waypoints.

The risk zones constitute the dynamic elements of the scenario, and they consist of pop-ups with a known future probability of appearance. For each pop-up the system generates at planning-time several feasible evasion manoeuvres, qualified by a set of route parameters (fuel consumption, assumed risk and spent time) used at flight-time as decision variables to optimize the route. The work extends our preliminary results on trajectory generation over static environment (Ruz et al., 2006), taking into account the knowledge about pop-ups in the trajectory design. The route planner that is proposed in this work allows the UAV to make a decision among several alternative routes considering both, current state of the UAV and probabilities of pop-up threats appearance in future time. The planning is carried out in three steps. First, an initial and optimal path planning is designed taking into account only the static elements of the scenario. Second, alternatives routes are calculated to bypass each pop-up zone having an appearance probability greater than zero. Finally, these alternatives are attached to the original flying plan, and given to an upper layer module in charge of making decisions according to the imposed limitations of fuel, time, and risk.
The chapter is organized as follows. Section 2 presents the MILP techniques as a method for trajectory generation of UAVs. This section encompasses the formulation of constraints for UAV's dynamic, obstacle's avoidance, target reaching, radar's avoidance and target within radar zones. For the last set of constraints a linear approximation with indicator 0-1 variables is introduced so that the non-linear and non-separable terms of the radar detection function could be linearized. Section 3 describes the path planning approach based on a modification of the A* algorithm. Section 4 explains how the planner presented in this work modifies the optimal trajectory to avoid pop-up threats. This section examines the three parameters utilized in the election of an alternative route that bypasses the threats: mean risk, flying time and fuel, and poses the corresponding decision making as an Integer Linear Programming (ILP) problem. In section 5, we present the implementation and some results of the planner. Finally, section 6 presents the conclusions and future works.

2. MILP as a method for trajectory planning of UAVs

Mixed Integer Linear Programming (MILP) is a powerful mathematical programming framework that extends continuous linear programming to include binary or integer decision variables to encode logical constraints and discrete decisions together with the continuous vehicle dynamics. The approach to optimal path planning based on MILP was introduced in (Schouwenaars et al., 2001). The UAV's trajectory generation is faced as a 3D optimization problem under certain conditions (see Fig. 1) in the Euclidean space, characterized by a set of *decision variables*, a set of *constraints* and the *objective function*. The *decision variables* are the UAV's state variables, i.e. position and speed. The *constraints* are derived from a simplified model of the UAV and the environment where it has to fly on. These constraints include:

- Dynamics constraints, such as a maximum turning force which causes a minimum turning radius(C_R), as well as a maximum climbing rate (C_C).
- Obstacles avoidance constraints like no-flight zones (C_O)
- Target reaching constraints of a specific waypoint or target (C_T).
- Radar avoidance constraints (C_R)

The *objective function* includes different measures of the quality in the solution of this problem, although the most important criterion is the minimization of the total flying time

to reach the target. However, when there are threats that put in risk the UAV mission, for example missiles guided by radars, it becomes necessary to incorporate to the objective function some term that minimizes the risk of UAV detection (R_D). In this work the MILP approach for path planning is expanded to include linear constraints derived from a non-linear risk detection model of the UAV by radars. The MILP alternative to plan a static route will be mainly used when the environment contains threats that put under certain risk the UAV's mission, for instance the tracking and the shooting of missiles guided by radars.

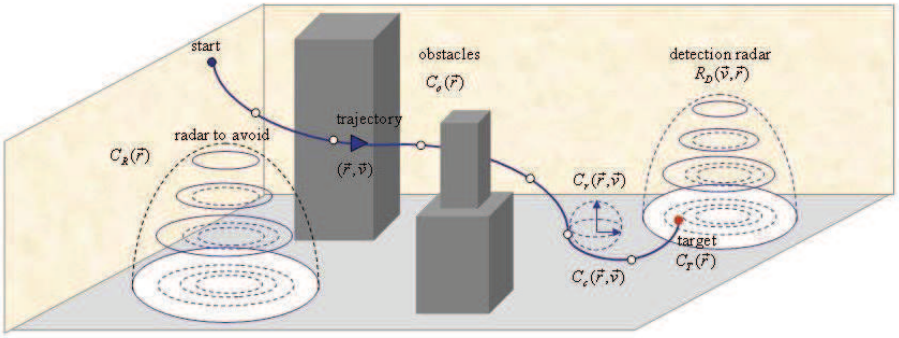


Figure 1. Trajectory generation as a MILP constrained optimization problem

2.1 Dynamic constraints

The trajectory optimization is constrained by a time-discrete dynamics to obtain the state of the system on every time t + 1, from its previous state at time t. This model represents the UAV with its limits in speed, climbing rate and turning rate, caused by a maximum magnitude in the input force u(t) that the aircraft might be put under (Bellingham & How, 2002).

Therefore the dynamics is expressed in (1).

$$\begin{bmatrix} \vec{v}(t+1) \\ \vec{r}(t+1) \end{bmatrix} = A \cdot \begin{bmatrix} \vec{v}(t) \\ \vec{r}(t) \end{bmatrix} + B \cdot \vec{u}(t) \tag{1}$$

The matrices A and B are the state matrices (kinematics matrices) of the UAV, and t the discrete time. Speed and force magnitudes are constrained by the following equations which are convenient linearizations of the original quadratic form:

$$\begin{aligned} & v_x \cos(\varphi_n)\sin(\theta_m) + v_y \sin(\varphi_n)\sin(\theta_m) + v_z \cos(\theta_m) \le |\vec{v}_{\max}| \\ & u_x \cos(\varphi_n)\sin(\theta_m) + u_y \sin(\varphi_n)\sin(\theta_m) + u_z \cos(\theta_m) \le |\vec{u}_{\max}| \\ & \varphi_n = 2\pi n / N; \qquad \theta_m = \pi m / M \\ & n = 1,2,3,...N; \qquad m = 1,2,3,...M \end{aligned} \tag{2}$$

where the maximum speed and input force are in the right side of (2). N and M are the number of points used to approximate the spherical space closed by all the possible directions of a vector in 3D (Schouwenaars et al., 2004). The angle θ is the zenith and the angle φ is the azimuth of vector $\vec{u}$ represented in spherical coordinates.

Additionally, in order to model the climbing rate of a UAV, the components of speed and acceleration in the Z axis must be constrained according to its manoeuvring capabilities (3).

$$\begin{aligned} &\forall t \\ &v_{z\_\min} \le v_z(t) \le v_{z\_\max} \\ &u_{z\_\min} \le u_z(t) \le u_{z\_\max} \end{aligned} \tag{3}$$

At the same time the third component of the UAV's position must be limited to one side of the Z axis, which in our case is the positive one, because the implemented referential frame has been the ENU (East-North-Up → {x,y,z}).

2.2 Obstacles avoidance constraints

As mentioned in the beginning of this section, the optimization considers box obstacles avoidance, which is modelled by knowing the lower south west and the upper north east vertexes of the no-flight zones (Kuwata & How, 2004), as expressed by the constraints in (4):

$$\begin{aligned} x_i(t) &\le x_{i\min}^k + M_o \cdot \delta_j^k(t) \\ x_i(t) &\ge x_{i\max}^k - M_o \cdot \delta_j^k(t) \end{aligned} \tag{4}$$

where k identifies the obstacle, (x_1min,x_2min,x_3min) is its lower south west corner and (x_1max,x_2max,x_3max) its upper north east corner, M_o is a bound for $x_i(t)$ to enable/disable some of the constraints, and δ_j^k are the indicator variables. The jth condition is then relaxed when $\delta_j^k = 1$, and enabled when $\delta_j^k = 0$. By (5) it is imposed that at least one of these constraints is active.

$$\begin{aligned} &\sum_{j=1}^{6} \delta_j^k(t) \le 5 \\ &\forall k \;/\; k = 1,2,...,K_{obst} \;\wedge\; \forall t \;/\; t = 0,1,...,T_{\max} \end{aligned} \tag{5}$$

2.3 Target reaching constraints

The constraints in (6) must be satisfied for all instant during the flight, to impose the UAV to get to its final destination, such as a waypoint or a target.

$$\begin{aligned} &x_i(t) - x_{i_f} \le M(1-\lambda_t) \\ &x_i(t) - x_{i_f} \ge -M(1-\lambda_t) \\ &\sum_{0}^{T} t \cdot \lambda_t \le t_{arrival}, \quad \forall t \end{aligned} \tag{6}$$

In the arrival constraints, (x_f,y_f,z_f) is the target location, λ_t is a binary indicator variable to enable/disable each constraint, and $t_{arrival}$ is the time to be included as a term of the objective function to be minimized (Ruz et al., 2006).

2.4 Radars avoidance constraints

The treatment developed to avoid threat zones where there are enemy units such as tracking radars and missile launchers is based on a distribution which corresponds to the probability

for the UAV to be detected, tracked and shot. This probability distribution is characteristic for every single type of tracking radar (7), where c_1 and c_2 are constants obtained by a curve fitting routine. This fit is previously made using the information provided by the radar's specifications.

$$P_t = \frac{1}{1 + c_2 \left(R^4 / \sigma \right)^{c_1}} \tag{7}$$

In the tracking probability distribution the value σ represents the RCS (radars cross section) in squared meters, while the variable R is the distance in meters between the UAV and the radars. In Fig. 2 it is observed that a radius equal to 50Km could be considered as the maximum radar's range, from which P_t totally diminishes.

Thus, for any maximum accepted value of detection, tracking, or killing probability its corresponding minimum radius $R_{\min}$ to avoid the radar's zones is computed. This avoidance condition is then expressed in (8), where a linearization of the quadratic form of all the possible directions for a vector in the Euclidean space is used once again to define the constraints in the MILP model.

$$\begin{aligned} & x_R \cos(\varphi_n)\sin(\theta_m) + y_R \sin(\varphi_n)\sin(\theta_m) + z_R \cos(\theta_m) \geq \left| \vec{R}_{\min} \right| \\ & \varphi_n = 2\pi n / N; \;\; \theta_m = \pi m / M; \;\; n = 1,2,...N; \;\; m = 1,2,...M \end{aligned} \tag{8}$$

The coordinates x, y and z are the relative coordinates of the UAV to each radar, making the aircraft able to approximate then differently, depending on the types of radars and on the interests in any mission in particular.

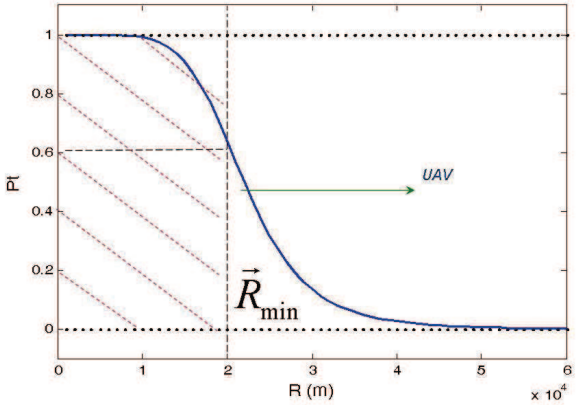


Figure 2. Profile of a tracking probability distribution with a RCS = 1.0m²

2.5 Constraints for targets within radar zone

When the UAV must reach a target within a radar zone, the detection risk must be minimized. To address such a situation we have used a model based on idealizing geometrical and physical UAV properties (Murphey et al., 2003). The model assumes that in the radar zone the UAV approaches to the target holding a fixed height (h). The detection

risk D is proportional to the UAV's RCS, σ, reciprocal to the fourth power of the distance between the UAV and the radar, r_R and independent of the UAV speed (Figure 3).

$$D = \sigma / r_R^4 \tag{9}$$

The UAV is considered to be an ellipsoid with the axis of symmetry determining the direction of the UAV trajectory (Murphey et al., 2003).

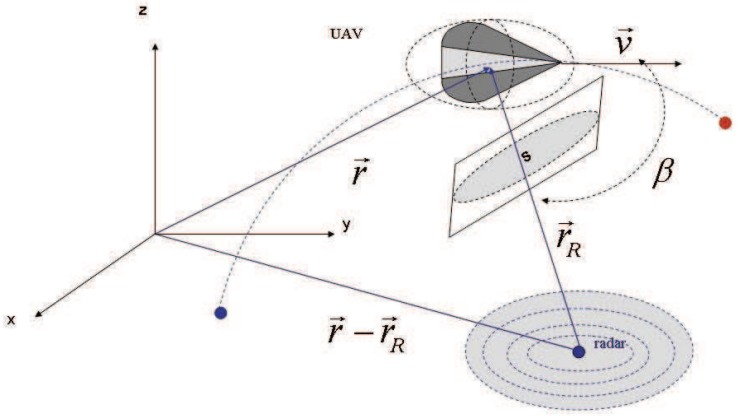


Figure 3. Layout of an UAV under radar detection risk

The UAV's *RCS* exposed to the radar is proportional to the area of the ellipsoid's projection onto the plane orthogonal to vector $\vec{r}_R$, as shown in Figure 3. Therefore, the *RCS* is given by:

$$\sigma = kS \tag{10}$$

Where k depends on the radar technical characteristics, and S is the ellipsoid's projection given by Eqn.11.

$$S = \pi\sqrt{a^2 \sin^2\beta + b^2 \cos^2\beta} \tag{11}$$

β is the angle between vectors $\vec{r}_R$ and $\vec{v}$ (velocity), and the ellipsoid is defined by semi-axis a and b. As the altitude is fixed (h), β will be a function of x, y, v_x, and v_y, then the detection risk might be expressed as in Eqn.12, such that:

$$D(x, y, v_x, v_y) = k\sqrt{\frac{(hv_y)^2 + (hv_x)^2 + (v_x y - v_y x)^2\left(\frac{b^2}{a^2}\right)(v_x x + v_y y)^2}{(v_x^2 + v_y^2)(x^2 + y^2 + h^2)^9}} \tag{12}$$

In order to include the detection risk function (Eqn.12) in a MILP framework, it has been approximated by a piecewise first order linearization D_L showed in Eqn.13.

$$D_L = \sum_i (a_i x(t) + b_i y(t) + c_i v_x(t) + d_i v_y(t) + e_i)\delta_i(t) \tag{13}$$

where a_i, b_i, c_i, d_i, and e_i are the coefficients of the i^{th} hyper-plane inside the i^{th} domain, $\delta_i(t)$ is a binary indicator variable identifying the domain $[lx_i, ly_i, v_x, v_y]$ -$[ux_i, ux_i, v_x, v_y]$ that must take value 1 only in this domain, hence it must be constrained to:

$$\begin{aligned} &\sum_{i=1}^{p} lx_i \delta_i \le x(t) \le \sum_{i=1}^{p} ux_i \delta_i \\ &\sum_{i=1}^{q} ly_i \delta_i \le y(t) \le \sum_{i=1}^{q} uy_i \delta_i \\ &\sum_{i=1}^{p \cdot q} \delta_i(t) = 1 \end{aligned} \tag{14}$$

To change the positive and negative range of $x(t)$, $y(t)$, $v_x(t)$, $v_y(t)$ to the [0, 1] interval, we apply the following change of variables:

$$\begin{aligned} &x(t) = x^b(t)[u_x - l_x] + l_x \\ &y(t) = y^b(t)[u_y - l_y] + l_y \\ &v_x(t) = v_x^b(t)[u_{vx} - l_{vx}] + l_{vx} \\ &v_y(t) = v_y^b(t)[u_{vy} - l_{vy}] + l_{vy} \\ &0 \le x^b(t), x^b(t), v_x^b(t), v_y^b(t) \le 1 \end{aligned} \tag{15}$$

where u and l are upper bounds and lower bound of x, y, v_x and v_y , respectively. To linearize the products $V^b(t)\delta_i(t)$ (where $V^b(t)= x^b(t), y^b(t), v^b(t), v^b(t)$) of a binary variable $\delta_i(t)$ and a continuous variable $V^b(t)$, we replace $V^b(t)\delta_i(t)$ for

$$W_i^b(t) = x_i^b(t), y_i^b(t), v_{x,i}^b(t), v_{y,i}^b(t) \tag{16}$$

and impose the following constraints.

$$\begin{aligned} &W_i^b(t) \le \delta_i(t) \\ &W_i^b(t) \le V_i^b(t) \\ &W_i^b(t) \ge V_i^b(t) + \delta_i(t) + 1 \end{aligned} \tag{17}$$

So, piecewise detection risk function will be:

$$\begin{aligned} &D_L = \sum_k A_k x_k^b + B_k y_k^b + C_k v_{x,k}^b + D_{ij} v_{y,k}^b + E_k \delta_k \\ &where \quad A_k = a_{ij}(u_x - l_x), \ldots \end{aligned} \tag{18}$$

The coefficients are calculated by selecting five points over the hyper-surface of D, and solving the corresponding system of equations. Fig. 4 shows the original function D and the

piecewise approximation D_L. The relative error of D achieved with this approximation was, for the implemented number of linearization, under 5%.

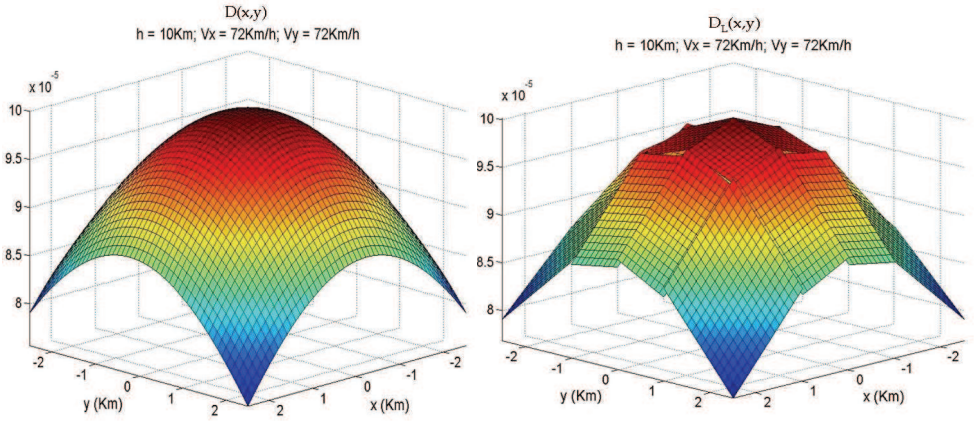


Figure 4. Radar detection function D(x,y) and its linearized radar function D_L(x,y)

3. Trajectory generation using A*

It is well known that the A* is a general AI search algorithm which is highly competitive with other path finding algorithms, and yet very easy to implement (Melchior et al., 2003; Szczerba et al., 2000; Trovato, 1996) . For UAV's trajectory generation it has the advantage to present a good performance when the terrain is quite irregular and the manoeuvres must be done close to the ground's surface. The main drawback is its high computational cost in 3D path planning problems, but it is reasonable to deal with this fact when an appropriate scaling is done. In the case under study the A* algorithm has been modified to adapt it and improve its capabilities to optimize trajectories for UAVs under conditions like no-flight zones, scarped terrains, and radars, also called ADUs (air defence units) which might appear dynamically during the flight. The modifications done to the original A* algorithm are mainly logical restrictions which evaluate whether the successors of every possible node of the UAV's trajectory build a minimum turning radius curve, while avoiding physical obstacles and threats.

3.1 Flying direction constraint

The constraint imposed on the flying direction is the computation of the angle between the arriving direction and every possible departing direction, for all the nodes of the trajectory. The angle ϕ is calculated supported by the definition of the dot product (19).

$$\phi = arc\cos\left(\frac{\vec{V}_f \cdot \vec{V}_i}{\left|\vec{V}_f\right| \cdot \left|\vec{V}_i\right|}\right) \tag{19}$$

Only successors with angle values less than a threshold (set to $\pi/2$ in this work) are allowed during the expansion. Fig. 5 shows a 2D view of the space explored by the A* algorithm when put under this condition.

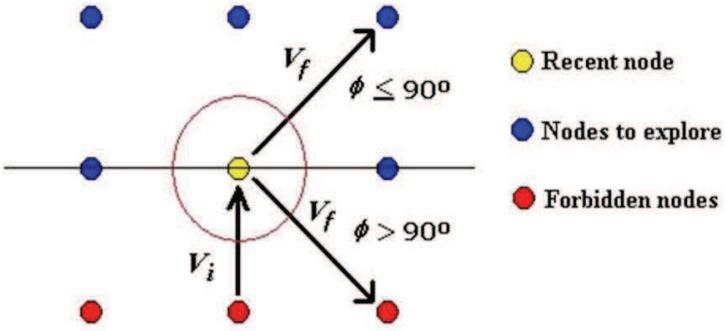


Figure 5. Nodes explored by the 3D A* algorithm

This modification not only intends to emulate the fixed wings aircraft limitation to turn immediately around when flying at high speeds, but also reduces the computing time because the algorithm expands less nodes than original version.

3.2 UAV's inertia constraint

The existence of waypoints in the designed missions leads us to impose a set of conditions of continuity on them, for the kinematical expressions of the UAV (20), and over all the passages from one section to the next one.

$$\begin{aligned} &r(t+\Delta t) \rightarrow r(t); \\ &v(t+\Delta t) \rightarrow v(t); \\ &a(t+\Delta t) \rightarrow a(t); \\ &\forall t \wedge \Delta t \rightarrow 0 \end{aligned} \tag{20}$$

This is modelled within the A* algorithm by constraining the first node to expand from each waypoint according to the initial conditions. Thus, the initial flying direction is given to the algorithm so it can project it onto all the possible directions towards the expanded nodes.
The projection with the highest value is selected, and determines the next node to be incorporated to the trajectory. This modification on the algorithm tries to model the UAV's inertia, and the behavior of the resulting trajectory depends on the A* resolution, which should be properly calibrated with the size of the scenario and the UAV's speed.

3.3 Refinement of the A* space

The resolution of the A* space is related to the ratio between the total number of nodes and the size of the flying space. For example, a defined longitudinal density ρ_i along the i-th axis will be the total number of nodes N_i chosen to represent that real dimension into the A* space, over the total length L_i of the scenario on this axis (21).

$$\rho_i = \frac{N_i}{L_i} \tag{21}$$

From this definition it is possible to infer that the more the number of nodes in the A* space the higher the resolution of the algorithm. However, there is an important limitation for the selected resolution, because this one carries a combinatory of the order ~N_i^3 when looking for the optimum path. For spaces with approximately 100 nodes per dimension this results in a high computational cost. But a lower resolution might be not enough when used to represent a space with surface irregularities. To save computing time without loosing information of the terrain we have carried out a fine sampling between each pair of nodes expanded by the algorithm, as seen in Fig. 6.

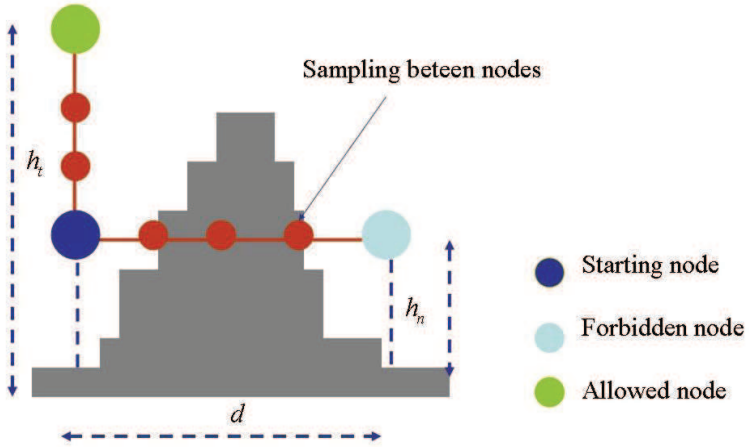


Figure 6. Improving the resolution with a finer sampling between nodes of the A* space

The sampling introduces a combinatory of the order ~N_i^2 to the searching task of the algorithm that, thanks to this modification only looks for the coordinates of the space in the *X*-*Y* plane, and compares the sampling point's height with the altitude of the surface, to discriminate nodes that could hide abrupt changes of the soil's profile, like cliffs or mountain peaks. In this way the algorithm remains computationally bearable while the surface is included in the path planning with an acceptable resolution.

4. Modification of trajectories in dynamic environments

The scenario under study is made up of one UAV, no-flight zones, obstacles (buildings), terrain, fixed ADUs, and pop-up ADUs as dynamic elements. The uncertain appearance of these last elements encourages the search of a routine to optimize decisions made during planning time and, if possible, on line, depending on how the mission is evolving in time. Fig. 7 shows the way the UAV updates information about previously known dynamic elements and re-launch the decision making ILP routine if needed, or even discovers new pop-ups and, if it has enough computing resources, re-executes the planning from the bypassing up to the decision making process, trying to follow again the main trajectory.

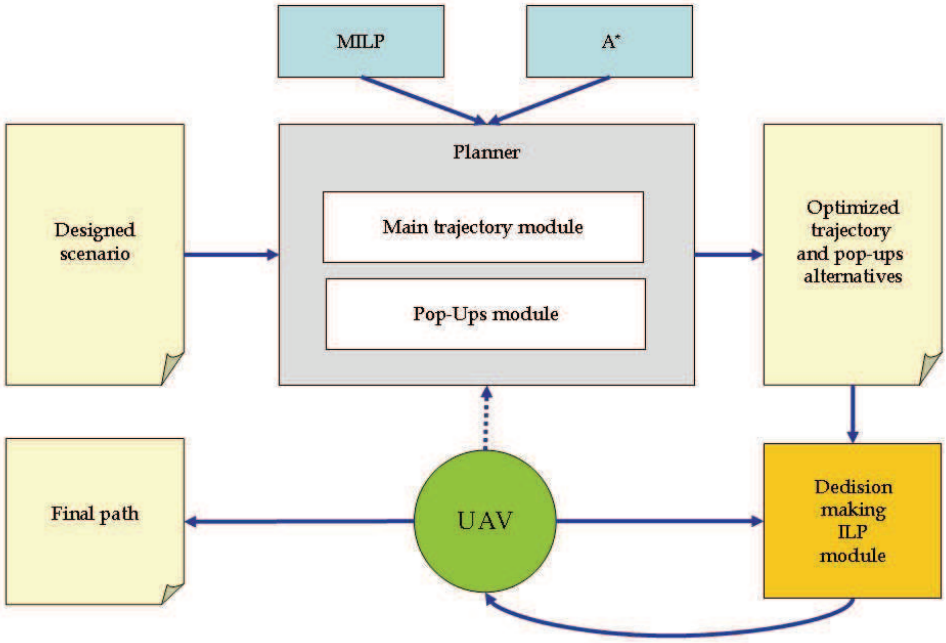


Figure 7. Layout of the path generator

The original trajectory is modified according to a decision which is made based on the following three elements: mean risk, flying time, and fuel consumption. The ILP formulation computes the optimum alternatives to be chosen (Ruz et al., 2007).

4.1 Cumulative mean risk parameter

We start from the computation of the kill probability (P_K) on each point of a trajectory, from which a measure of the cumulative mean risk for that particular trajectory is computed. This computation of the probability of kill can be made by any available analytical, statistical, or even empirical model of the interactions between the aircraft and the enemy defence units, as seen before in (7). It is also possible to count with tabulated values of P_K, for several ranges of the variables that affect this probability such as UAV's position and RCS.
From the kill probability P_{Ki} at any trajectory point, the survival probability P_{Si} associated to the enemy *i*-th ADU is computed as follows in (22).

$$P_{Si}(\vec{r},\vec{v}) = 1 - P_{Ki}(\vec{r},\vec{v}) \tag{22}$$

The total survival probability P_{ST} is computed by multiplying the following three factors (23):

1. The P_{Si} product from 1 to the number N of known ADUs.
2. The Pop-Up ADU survival probability affecting the point, P_{SPU}.

3. A weighting factor α embedding the Pop-Up appearance probability when flying towards its location.

We have chosen the product combination based in the assumption that all the ADUs work in cooperation.

$$P_{ST}(\vec{r},\vec{v}) = \alpha P_{SPU}(\vec{r},\vec{v}) \prod_{i=1}^{N} P_{Si}(\vec{r},\vec{v}) \tag{23}$$

The weighting factor α, ranging in (0, 1), is defined as the complementary probability of the pop-up appearance probability P_{APU}, at any state $(\vec{r},\vec{v})$ of the aircraft (24). It should be strictly greater than zero and less than one, because the fact to exactly assign any of these values to a P_{APU} probability makes the ADU turn into a nonexistent or fixed one respectively, which is already considered in the last term of the product.

$$\alpha = 1 - P_{APU} \tag{24}$$

Then, it is clear that the more probable a pop-up arises in front of the UAV during the flight the less probable the UAV will survive to it, bringing a greater cumulative mean risk to the chosen trajectory.

Once the total survival probability to a set of N cooperating ADUs is computed, including the unexpected activation of more threats, the total probability of kill P_{KTm} in the state $m = (\vec{r},\vec{v})$ is given by the complement to one (25).

$$P_{KTm}(\vec{r},\vec{v}) = 1 - P_{STm}(\vec{r},\vec{v}) \tag{25}$$

Hence, we define the cumulative mean risk of a trajectory R_K as the average of the total kill probabilities of all the M points which form this trajectory (26). This concept will be used as a parameter to characterize the group of alternatives to build a final trajectory under a decision making formulation.

$$R_K = \frac{1}{M} \sum_{m=1}^{M} P_{KTm}(\vec{r},\vec{v}) \tag{26}$$

The risk is calculated as a mean value, based on the discrete time system assumption. The points of the trajectory are approximately equally spaced since the flying speed is constant. If the time were continuous the integral form to calculate a mean value would be used instead of the sum showed in (26).

4.2 Cumulative flying time parameter

The flying time parameter is simply a way to characterize alternative trajectories in terms of the cumulative time needed by the aircraft to achieve them, assuming a constant flying speed. Thus, for an M-points time discrete path, with all points equally spaced in Δt time, the total flying time T is given by (27).

$$T = \sum_{m=1}^{M} \Delta t_m = M \Delta t \tag{27}$$

There is also a way to normalize the cumulative time parameter, with the aim to compare different alternatives of a trajectory or even different trajectories. If we define the amount τ as the time to go along the minimum length/time trajectory between any pair of points (straight line), it is possible to define a normalized cumulative flying time factor f_t (28), where the zero value represents a characterization for the mentioned minimum path.

$$f_t = 1 - \frac{\tau}{T} \tag{28}$$

4.3 Cumulative fuel factor parameter

Since the UAV's trajectory generation is represented as a 3D optimization problem, it might be formulated with an objective function and a set of constraints in a Cartesian referential frame where (x,y,z) is the UAV's position. Among the constraints there are kinematical and dynamical limitations of the system, which is an air vehicle unable to make stationary flights. Furthermore, the linear approach and the time discrete character of the solution led us to the matrix representation already shown in (1), with the limits that produce a minimum turning radius possible to achieve.

A more convenient expression for the limits of speed and acceleration as a function of their components in $\boldsymbol{R}^3$ is in (29), where again the maximum limits are in the right side of the inequalities.

$$\begin{aligned} v_x{}^2 + v_y{}^2 + v_z{}^2 &\le \left|\vec{v}_{\max}\right|^2 \\ u_x{}^2 + u_y{}^2 + u_z{}^2 &\le \left|\vec{u}_{\max}\right|^2 \end{aligned} \tag{29}$$

It is possible to reorder the constraint for the maximum turning rate making normalization for each of the acceleration's components (30), where the angle θ is the zenith and the angle φ is the azimuth of vector $\vec{u}$ represented in spherical coordinates.

$$\frac{u_x \cos(\varphi)\sin(\theta)}{\left|\vec{u}_{\max}\right|} + \frac{u_y \sin(\varphi)\sin(\theta)}{\left|\vec{u}_{\max}\right|} + \frac{u_z \cos(\theta)}{\left|\vec{u}_{\max}\right|} = C_t \tag{30}$$

Thus, in (31) it is shown the constraint for the signal C_t, which is a normalized input signal for each discrete time t. It represents a way to measure the acceleration applied to the system, needed to change the flying direction at any time step. This signal can be considered directly bounded to the aircraft fuel consumption because it might be the control signal, or the actuator signal used to change the UAV's course.

$$C_t \le 1 \qquad \forall t \tag{31}$$

Finally, in (32) we define the fuel consumption factor as the average of the fuel consumed along the t trajectory points.

$$F_c = \frac{1}{T}\sum_{t=1}^{T} C_t \tag{32}$$

4.4 Decision making module

In every mission the path designers might count with very accurate information about most of the elements involved in the flying environment, which can be provided and confirmed by several sources during the planning time. However, it is also possible to possess a minimum knowledge about uncertain or dynamic elements characterized by a probability of appearance, and that might represent a threat for the UAV's path.

The strategy proposed in this work implements an initial path planning taking into account only the well known and fixed components of the scenario, to obtain the main optimum trajectory which will be followed by the UAV. After having a main route, the knowledge of non-static elements, such as pop-up radars, is included in the scenario for considering only those pop-ups that actually may be a serious threat for the UAV. Once the actual threats have been discriminated from all the originally counted, a local avoidance strategy is computed, using MILP or A* algorithm, to bypass the pop-ups. These alternatives are all attached to the original flying plan, and given to an upper layer module in charge of making decisions according to the imposed limitations; let's say fuel consumption, time, and risk. It is right here where an optimum decision making process will increase the chances of a successful mission.

Suppose there is a mission to go from a starting point to an objective, as seen in figure 8, and that the originally planned trajectory might be affected by three independent unforeseen threats, characterized by their corresponding appearance probability P_{PU}. Therefore, each of them has an associated probability factor α of nonappearance, which assigns certain weight to the survival probability of the aircraft against those pop-ups.

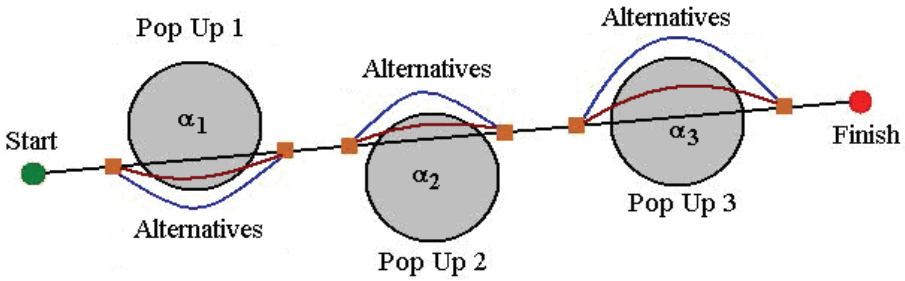


Figure 8. Trajectory decision map with three possible pop-ups and the corresponding three alternatives to avoid each of them

If the number n_a of alternatives is the same for each pop-up in particular, it's easy to compute the total amount of combined alternatives. In this case, the combinatory leads us to a total amount of alternatives (n_a^{Npu}), which is the number of alternatives by pop-up powered to the total number of pop-ups N_{pu}.

All the alternatives have their characteristic parameters to be processed in a decision making algorithm that seeks and find the optimum final trajectory, based not only on the recent and past information at the moment of the decision, but also on the probability of future events.

The choice of the optimal sequence of alternatives that will compose the final planned route can be posed as an ILP problem. The cumulative time and fuel consumption parameters will be the constraints, and the cumulative mean risk the objective (minimum) function. The mentioned objective function is given by (33), where L is the total number of pop-ups

affecting the original trajectory (pre-planned trajectory without pop-ups), and the indexes {*i,j,…,w*} range over all the alternatives for each one of the affecting pop-ups.

$$J = \left(\sum_i R_{K1i}\delta_{1i} + \sum_j R_{K2j}\delta_{2j} + \ldots + \sum_w R_{KLw}\delta_{Lw} \right) \tag{33}$$

In this objective function the coefficients R_{Klm} are the cumulated mean risk of each alternative, and the variables δ_{lm} binary variables associated to the chosen alternative among all the possible ones for each pop-up. Therefore, the variables must be constrained (34), to guarantee that only one of the alternatives is selected at the time of making a specific decision.

$$\sum_{i=1}^{I} \delta_{1i} = 1; \ \sum_{j=1}^{J} \delta_{2j} = 1; \ \ldots; \ \sum_{w=1}^{W} \delta_{Lw} = 1 \tag{34}$$

The rest of the constraints refer to the upper limit assigned to the accepted cumulative time factor (35), and to the maximum cumulative fuel consumption factor (36). Both limits can be set based on the UAV's dynamics, and on its fuel consumption model.

$$\frac{\left(\sum_i T_{1i}\delta_{1i} + \sum_j T_{2j}\delta_{2j} + \ldots + \sum_w T_{Lw}\delta_{Lw} \right)}{L} \leq T_{\max} \tag{35}$$

The T_{lm} coefficients are the cumulative time factor of every computed alternative, and $T_{\max}$ is the limit accepted for the time factor of the mission.

$$\frac{\left(\sum_i F_{c1i}\delta_{1i} + \sum_j F_{c2j}\delta_{2j} + \ldots + \sum_w F_{cLw}\delta_{Lw} \right)}{L} \leq F_{c\max} \tag{36}$$

The coefficients F_{clm} are the cumulated fuel consumption factor of every computed alternative, and $F_{c\max}$ is the upper limit for the mean fuel consumption along the global trajectory.

5. Implementations and results

A path planning software platform was developed implementing both, MILP and A* algorithm trajectory optimizers. The MILP model takes advantage of the powerful CPLEX 9.0 solver through the ILOG CPLEX package (ILOG, 2003), to find the solution for optimum trajectories in the space of the discrete UAV's variables of state. The A* algorithm was coded in JAVA language, using the JRE system library jre1.5.0_06. The metric used as the heuristic was the Euclidean distance.

Figure 9 shows the resulting trajectory computed in a scenario where there are mountains, waypoints, and pop-up radars only. The black solid line would be the optimum path whenever the pop-ups don't get enabled during the UAV's approximation. The yellow

dashed line is an alternative calculated during the planning time to safely escape from the threat that possibly causes a mission fail.

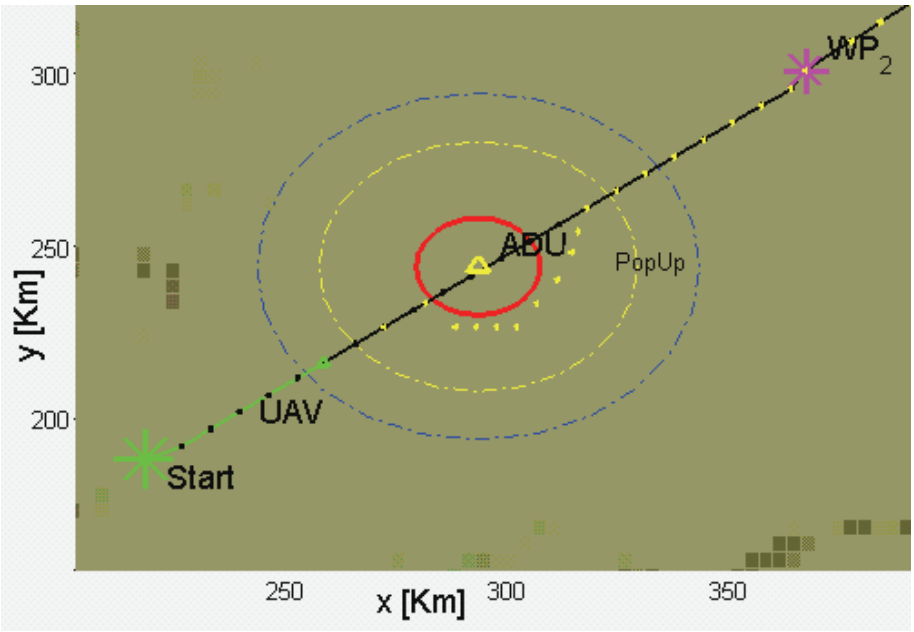


Figure 9. Computed trajectory (black) with the alternative (yellow) to avoid one pop-up

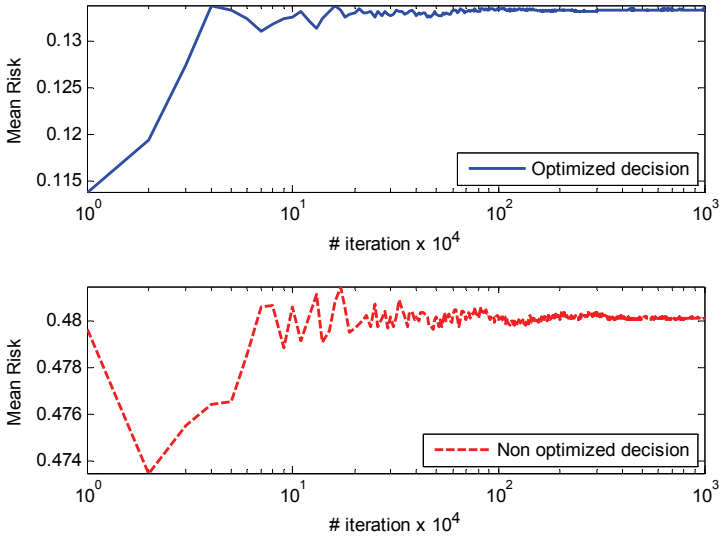


Figure 10. Cumulative mean risk after 10^7 Monte Carlo simulations

A Monte Carlo simulation was done to evaluate the decision making strategy proposed in this work (Berg & Chain, 2004), where a probability of future appearance assigned to every threat pop-up is taken into account to activate them, while the parameters of risk, time and fuel are constrained in an ILP model. This strategy was compared with the simple decision made on the basis of the consumed fuel and the spent time, which are only past and present sources of information. Figure 10 shows the cumulative mean risk of both strategies, after 10^7 iterations, where there were three pop-ups, with three alternative trajectories each. The probabilities of appearance were 0.5, 0.2, and 0.8, for the pop-ups affecting the original trajectory in that chronological order. As mentioned in section 1 these probabilities are provided by expert knowledge prior to the mission design. Depending on the selected alternative the mean risk accumulated different values. The more direct is the route, the more risky it is, while the less time it spends. The greater the turning radius of the route is, the less the fuel is consumed. It might be possible to find trajectories with the maximum time spent along it without having the higher fuel consumption. Constraints over the time factor (0.35) and the fuel consumption (0.40) were imposed into the ILP decision making model, to obtain the optimum final global path.

The histograms in figure 11 of the two simulated strategies show the advantages of choosing the optimum decision plan, because the constraints over spent time and fuel consumption are never violated, while the cumulated risk in minimized. The strategy that only considers past and present information doesn't violate the time and fuel criteria either, but its response to the cumulated risk is very poor because the most probable pop-ups is not the necessary the first one to appear.

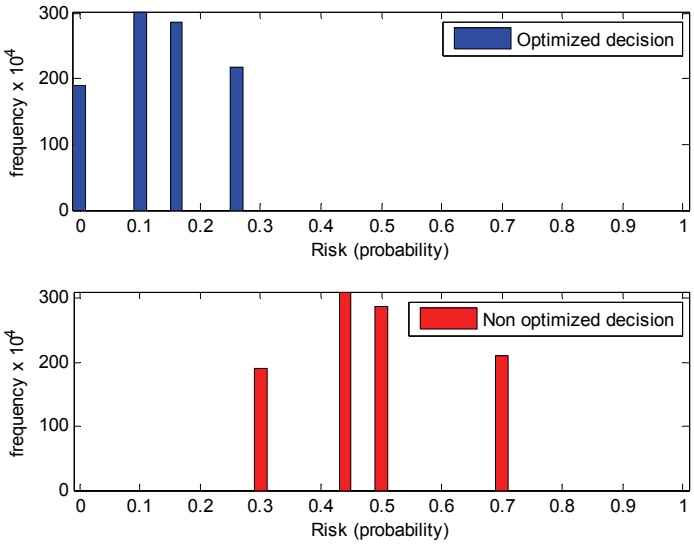


Figure 11. Histogram of the mean risk after 10^7 Monte Carlo simulations

Finally, figure 12 shows the results when the UAV's trajectory must reach its target within a radar zone. The detection risk is minimized respect to objective function (37)

$$J = \mu_1 t_{arrival} + \mu_2 D(x, y, v_x, v_y) \tag{37}$$

where D is the nonlinear radar detection function, μ_1 and μ_2 are weights which consider the importance of flight time and acceptable threat concerning to a particular mission.

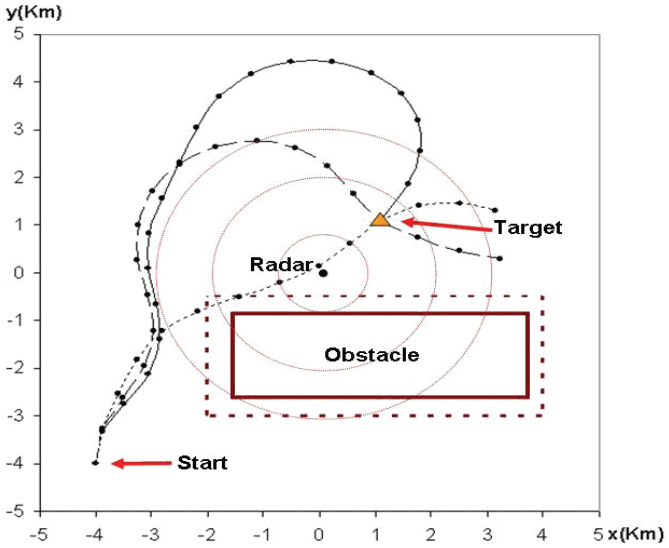


Figure 12. Comparison of three trajectories with target in radar zone

The UAV tries to avoid the radar detection by maintaining the biggest possible distance, compatible with the values μ_1 and μ_2, and controlling the RCS it presents to the radar. The trajectories plotted in Figure 12 shows that the UAV does not fly directly to the target, and when a higher risk of detection is even accepted, the UAV will use a more direct and risky trajectory (μ_1 = 1, μ_2 = 2.7e4). It can be observed that when the UAV is next to the target and the admitted risk is low (μ_1 = 1, μ_2 = 2.8e4), its trajectory tries to approach radially to the radar, minimizing its RCS. Over a no radar zone (μ_2 = 0) the flying trajectory goes directly to the target.

6. Conclusions and future work

We have presented the trajectory generation module of SPASAS, an integrated system for definition of flight scenarios, flight planning, simulation and graphic representation of the results developed at Complutense University of Madrid. The module uses two alternative methods, MILP and a modification of the A* algorithm, and considers static and dynamic environmental elements, particularly pop-ups. Both methods have been implemented and a Monte Carlo simulation was done to evaluate the decision making strategy proposed.

The results showed the advantages of choosing the optimum decision plan that considers the known values of the probability of appearance of pop-up threats in the future. The possibility to update the information concerning the pop-up's appearance probabilities, available fuel, time, and even the assumed risk, and then re-launch a decision making

routine to optimize the chosen alternatives has been proven, since the ILP model provides a solution affordable in real time (~1s).
When the UAV must reach a target within a radar zone, the detection risk is minimized using an efficient MILP formulation that approximates the continuous risk function with hyper-planes implemented with integer 0-1 variables.
For the future, we are already working in three main objectives: (a) the use of rotary-wing UAVs such as quad-rotors, (b) the introduction of video cameras onboard UAVs, and (c) the design of coordination algorithms for a fleet of UAVs. Rotary-wing UAVs will incorporate more maneuverability than conventional fixed-wing UAV, since they can take-off and land in limited space and can easily hover above a target. Cameras onboard will allow the use of vision-based techniques for locate and track dynamic perimeters as is needed in tasks such as oil spill identification or forest fires tracking. Finally, a team of UAVs will get an objective more efficiently and more effectively than a single UAV.

7. Acknowledgements

This research was funded by the Community of Madrid, project "COSICOLOGI" S-0505/DPI-0391, by the Spanish Ministry of Education and Science, project "Planning, simulation and control for cooperation of multiple UAVs and MAVs" DPI2006-15661-C02-01, and by EADS(CASA), project 353/2005.
The authors would like to thank Tomas Puche, Ricardo Salgado, Daniel Pinilla and Gemma Blasco from EADS(CASA), and Bonifacio Andrés, Segundo Esteban and José L. Risco from UCM, for their contribution to SPASAS project.

8. References

Bellingham, J. & How, J. (2002) Receding Horizon Control of Autonomous Aerial Vehicles. *Proc. of American Control Conference*.

Berg, B. & Chain, M. (2004) Monte Carlo Simulations and their Statistical Analysis. *World Scientific*, ISBN 981- 238-935-0.

Borto, S. (2000) Path planning for UAVs. *Proceedings of the American Control Conference*. pp. 364-368

How, J.; King E. & Kuwata, Y. (2004) Flight Demonstrations of Cooperative Control for UAV Teams. AIAA *3rd Unmanned Unlimited Technical Conference*, *Workshop and Exhibit.*

ILOG, Inc. ILOG CPLEX 9.1 (2003) User's guide, http://www.ilog.com/products/cplex.

Kuwata, Y. & How, J. (2004) Three Dimensional Receding Horizon Control for UAVs. *AIAA Guidance, Navigation, and Control Conference and Exhibit*.

Melchior, P.; Orsoni, B.; Lavialle O.; Poty A. & Oustaloup, A. (2003) Consideration of obstacle danger level in path planning using A* and fast-marching optimisation: comparative study. *Signal Process.* Vol. 83,11, pp. 2387-2396.

Murphey, R.; Uryasev, S. & Zabarankin, M. (2003) Trajectory Optimization in a Threat Environment. *Research Report 2003-9, Department of Industrial & Systems Engineering*. University of Florida.

Richards, A. & How, J. (2002) Aircraft Trajectory Planning with Collision Avoidance Using MILP. *Proceedings of the IEEE American Control Conference*. pp. 1936-1941.

Ruz, J.; Arévalo, O.; Cruz J. & Pajares, G. (2006) Using MILP for UAVs Trajectory Optimization under Radar Detection Risk. *Proc. of the 11th IEEE Conference on Emerging Technologies and Factory Automation*. ETFA'06, pp. 957-960.

Ruz, J.; Arevalo, O.; Pajares, G. & Cruz, J. (2007) Decision Making among Alternative Routes for UAVs in Dynamic Environments. *12th IEEE Conference on Emerging Technologies & Factory Automation*. ETFA'07,pp. 997-1004.

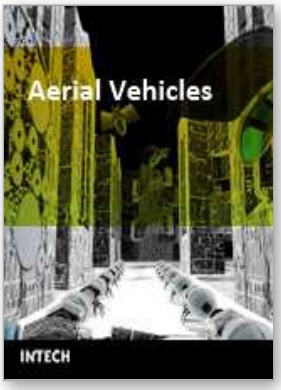
Schouwenaars, T.; Moor, B.; Feron, E. & How, J. (2001) Mixed Integer Programming for Multi-Vehicle Path Planning. *Proceedings of the European Control Conference*. pp. 2603-2608

Schouwenaars, T.; How, J. & Feron, E. (2004) Receding Horizon Path Planning with Implicit Safety Guarantees. *Proceedings of American Control Conference*. pp. 5576-5581.

Szczerba R.; Galkowski, P.; Glicktein, I. & Ternullo, N. (2000) Robust algorithm for real-time route planning. *IEEE Trans. Aerosp. Electron. Syst.* Vol. 36, 3, pp. 869-878.

Trovato, K. (1996) A* Planning in Discrete Configuration Spaces of Autonomous Systems. *PhD dissertation.* Amsterdam University.

Zengin, U. & Dogan, A. (2004) Probabilistic Trajectory Planning for UAVs in Dynamic Environments. *Proc. of AIAA 3rd Unmanned Unlimited Technical Conference, Workshop and Exhibit*. pp. 1-12.



Aerial Vehicles

Edited by Thanh Mung Lam

ISBN 978-953-7619-41-1
Hard cover, 320 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

This book contains 35 chapters written by experts in developing techniques for making aerial vehicles more intelligent, more reliable, more flexible in use, and safer in operation.It will also serve as an inspiration for further improvement of the design and application of aeral vehicles. The advanced techniques and research described here may also be applicable to other high-tech areas such as robotics, avionics, vetronics, and space.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jose J. Ruz, Orlando Arevalo, Gonzalo Pajares and Jesus M. de la Cruz (2009). UAV Trajectory Planning for Static and Dynamic Environments, Aerial Vehicles, Thanh Mung Lam (Ed.), ISBN: 978-953-7619-41-1, InTech, Available from: http://www.intechopen.com/books/aerial_vehicles/uav_trajectory_planning_for_static_and_dynamic_environments

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821