Path Planning for Cooperating Unmanned Vehicles over 3-D Terrain

Ioannis K. Nikolos and Nikos C. Tsourveloudis

Intelligent Systems and Robotics Laboratory, Department of Production Engineering and Management, Technical University of Crete, University Campus, 73100, Chania, Greece
jnikolo@dpem.tuc.gr, nikost@dpem.tuc.gr

Abstract. In this paper we suggest an off-line/on-line path planner for cooperating unmanned vehicles that takes into account the mission objectives and constraints through an optimization procedure. The cooperating vehicles can be either Unmanned Aerial Vehicles (UAVs) or Autonomous Underwater Vehicles (AUVs); these two categories of vehicles share common features as far as path planning is concerned and these features are used in this work for the development of a unified approach to the path planning problem over 3-D terrains. A number of unmanned vehicles of the same category are launched from the same or different known initial locations. The main issue is to produce 3-D trajectories (represented by 3-D B-Spline curves) that ensure a collision free path, respect the mission objectives and constraints, and guide the vehicles to a common final destination. The off-line planner is designed for known environments. The on-line one generates paths in unknown static environments, by exchanging acquired information from the cooperating vehicles' on-board sensors. For each vehicle a near optimum path is generated that guides it safely to an intermediate position within the already scanned area. The process is repeated for each vehicle until the final destination is reached by one or more members of the team. Then, each one of the remaining vehicles can either turn into the off-line mode to reach the target, moving through the already scanned area, or continue with the on-line mode. Both off-line and on-line path planning problems are formulated as optimization problems, and a Differential Evolution algorithm is used as the optimizer.

Keywords. 3-D Path Planning, Navigation, Vehicles Cooperation, UAVs, AUVs, Evolutionary Algorithms, Differential Evolution, B-Splines.

1 Introduction

Path planning is the generation of a space path between an initial location and the desired destination that has an optimal or near-optimal performance under specific constraints [1]. The main concerns during the comparison of various candidate solutions are *feasibility* and *optimality* [2]. Searching for optimality is not a trivial task and in most cases results in non-affordable computation time, even in simple problems. Therefore, in most cases we search for suboptimal or just feasible solutions.

In this work the path planning for cooperating unmanned vehicles moving over a 3-D terrain is considered; the vehicles can be either Unmanned Aerial Vehicles

(UAVs) or Autonomous Underwater Vehicles (AUVs). UAVs and AUVs share the common feature of performing inside a 3-D environment and having six degrees of freedom, although their kinematic characteristics are not the same. The upper ceiling for AUVs is the sea surface, while a similar upper ceiling exists for UAVs due to stealth considerations or flight envelop restrictions.

Path planning for UAVs and AUVs imply special characteristics that have to be considered [3], [4], [5], such as: (a) physical feasibility, (b) performance related to mission, (c) real-time implementation, (d) cooperation between the vehicles, (e) stealth (low observability due to the selected path). Besides their common features, differences also exist between the two categories, as far as coordination and path planning is concerned, which are mainly related with the different sensors and electronic equipment that are needed in order to cooperate and perform their mission.

Cooperation between robotic vehicles has gained recently an increased interest as systems of multiple vehicles engaged in cooperative behavior show specific benefits compared to a single one [6] [7].

Path planning problems are computationally demanding multi-objective multi-constraint optimization problems [8]. The problem complexity increases when multiple vehicles should be used. Various approaches have been reported for UAVs coordinated route planning, such as Voronoi diagrams [9], mixed integer linear programming [10], [11] and dynamic programming [12] formulations.

In Beard et al. [9] the motion-planning problem was decomposed into a waypoint path planner and a dynamic trajectory generator. The path-planning problem was solved via a Voronoi diagram and Eppstein's k-best paths algorithm, while the trajectory generator problem was solved via a real-time nonlinear filter.

In [13] the motion-planning problem for a limited resource of Mobile Sensor Agents (MSAs) is investigated, in an environment with a number of targets larger than the available MSAs. The problem is formulated as an optimization one, whose objective is to minimize the average time duration between two consecutive observations of each target.

Computational intelligence methods, such as Neural Networks [14], Fuzzy Logic [15] and Evolutionary Algorithms (EAs) [5], [16] have been successfully used to produce trajectories for guiding mobile robots in known, unknown or partially known environments. Besides their computational cost, EAs are considered as a viable candidate to solve path planning problems effectively; the reasons are their high robustness, their ease of implementation, and their high adaptability to different optimization problems, with or without constraints [16].

EAs have been successfully used in the past for the solution of the path-finding problem in ground based or sea surface navigation [17], [18], [19], or for solving the path-finding problem in a 3-D environment for underwater vehicles [20], [21].

Changwen Zheng et al. [5] proposed a route planner for UAVs, based on evolutionary computation. The generated routes enable the vehicles to arrive at their destination simultaneously by taking into account the exposure of UAVs to potential threats. The flight route consists of straight-line segments, connecting the way points from the starting to the goal points. The cost function penalizes the route length the high altitude flights or routes that come dangerously close to known ground threats.

In [22] a multi-task assignment problem for cooperating UAVs is formulated as a combinatorial optimization problem; a Genetic Algorithm is utilized for assigning the multiple agents to perform various tasks on multiple targets.

In [16] an EA based framework was utilized to design an off-line / on-line path planner for UAVs autonomous navigation. The path planner calculates a curved path line, represented using B-Spline curves in a 3-D terrain environment. The on-line planner gradually produces a smooth 3-D trajectory aiming at reaching a predetermined target in an unknown environment; the produced trajectory consists of smaller B-Spline curves smoothly connected to each other.

In this work the following scenario was considered: having a number of autonomous vehicles (either UAVs or AUVs), at the same or different known initial locations with predefined initial directions, we calculate 3-D smooth trajectories, which connect the initial locations with a single destination location, ensuring a collision free operation with respect to mission constraints. Each vehicle is assumed to be a point and its actual size is taken into account by equivalent obstacle growing.

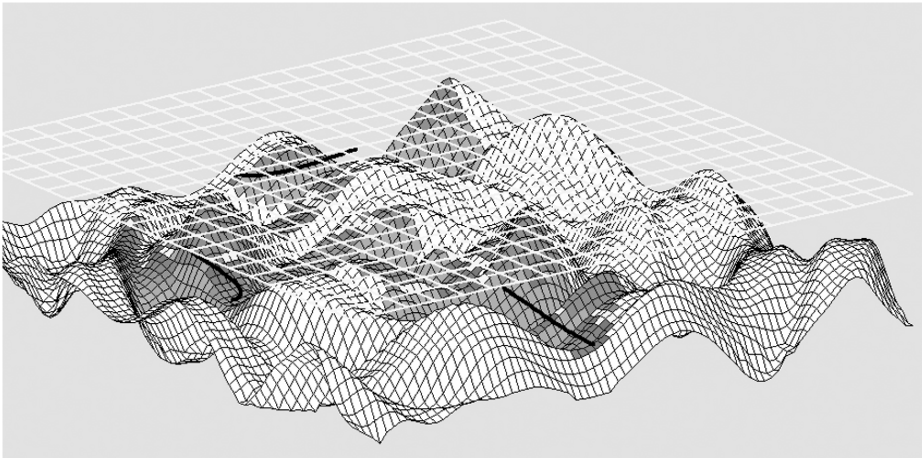

Fig. 1. A representation of the proposed concept: three vehicles are moving along curved path lines over a 3-D terrain; an upper ceiling is enforced (either sea surface or the maximum allowed flying height); on-board sensors are scanning the environment within a certain range in front of each vehicle.

Initially the off-line planner will be presented; it generates collision free paths in environments with known characteristics and flight restrictions. The on-line planner, being an extension of the off-line one, was developed to generate collision free paths in unknown environments. As each vehicle moves towards its destination, its on-board sensors are scanning the environment within a certain range and certain angles; this information is exchanged between the members of the team, resulting in a gradual mapping of the environment (Fig. 1). The on-line planner uses the acquired knowledge of the environment to generate a near optimum path for each vehicle that will guide it safely to an intermediate position within the known territory. The process is repeated until the corresponding final position is reached by one or more members of the team. Then, each one of the remaining members of the team either uses the off-line planner to compute a path that connects its current position and the final destination, or continues in the on-line mode until it reaches the common destination. Both path planning problems are formulated as minimization problems, where specially constructed functions take into account mission and cooperation objectives and constraints, with a Differential Evolution algorithm to serve as the optimizer.

The rest of the paper is organized as follows: in section 2 the off-line path planner for a single vehicle will be briefly discussed. Section 3 deals with the concept of on-line path planning for cooperating vehicles. The problem formulation is described, including assumptions, objectives, constraints, cost function definition and path modeling. Simulations results are presented in section 4, followed by discussion in section 5.

2 Off-Line Path Planner

The off-line planner generates collision free paths in environments with known characteristics and flight restrictions, where the solid boundaries are interpreted as 3-D surfaces. The derived path line for each vehicle is a single continuous 3-D B-Spline curve with fixed starting and ending control points. A third point, placed in a pre-specified distance from the starting one, is also fixed, determining the initial flight direction for the corresponding vehicle. Between the fixed control points, free-to-move control points determine the shape of the curve. For each path, the number of the free-to-move control points is user-defined.

Straight line segments that connect a number of way points have been used in the past to model UAV paths in 2D or 3D space [23], [5]. However, these simplified paths cannot be used for an accurate simulation of UAV's flight, unless a large number of way points is used. In [9], paths from the initial vehicle location to the target location are derived from a graph search of a Voronoi diagram that is constructed from the known threat locations. The resulting paths, consisting of line segments, are subsequently smoothed around each way point. Dubins [24] car formulation has been proposed as an alternative approach to the modeling of UAV dynamics [25]. This approach seems inefficient to model scenarios including 3D terrain avoidance and following of stealthy routes. However, this approach seems to be sufficient enough for task assignment purposes to cooperating UAVs flying at safe altitudes [13], [22], [25].

B-Spline curves have been used in the past for trajectory representation in 2-D [26] or in 3-D environments [16], [27]. They are well fitted in an optimization procedure as they need a few variables (the coordinates of their control points) to define complicated curved paths [28], [29]. The use of B-Spline curves for the determination of a path-line provides the advantage of describing complicated non-monotonic 3-dimensional curves with controlled smoothness with a small number of design parameters, i.e. the coordinates of the control points. Another valuable characteristic of the adopted B-Spline curves is that the curve is tangential to the control polygon at the starting and ending points. This characteristic can be used in order to define the starting or ending direction of the curve, by inserting an extra fixed point after the starting one, or before the ending control point.

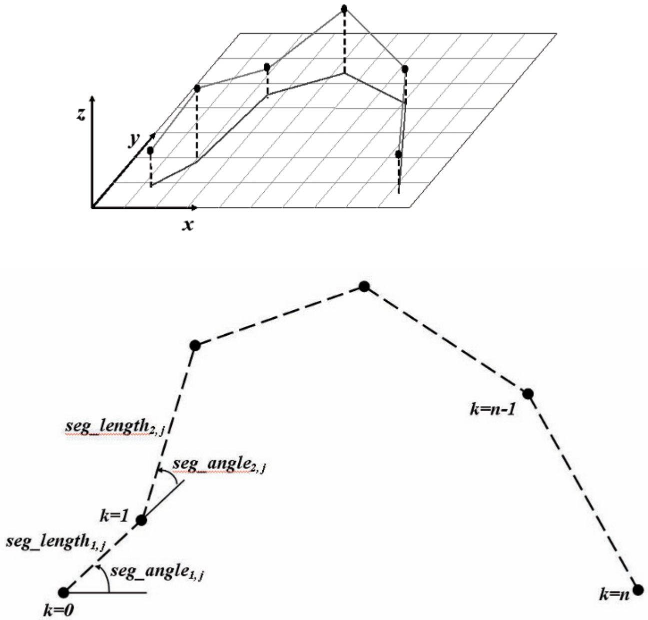

Fig. 2. Schematic representation of the B-Spline control polygon (top) and its projection on the horizontal plane (bottom).

In this work each path is constructed using a 3-D B-Spline curve; each B-Spline control point is defined by its three Cartesian coordinates $x_{k,j}$, $y_{k,j}$, $z_{k,j}$ (k=0,…,n, j=1,…,N, N being the number of vehicles, while n+1 is the number of control points in each B-Spline curve, the same for all curves). The first (k=0) and last (k=n) control points of the control polygon are the initial and target points of the j^{th} UAV, which are predefined by the user. The second (k=1) control point is positioned in a pre-specified distance from the first one, in a given altitude, and in a given direction, in order to define the initial direction of the corresponding path.

The control polygon of each B-Spline curve is defined by successive straight line segments (Fig. 2). Each segment of the control polygon is defined using its projection on the horizontal plane (Fig. 2); the length $seg\_length_{k,j}$, and the direction $seg\_angle_{k,j}$ of this projection are used as design variables (k=2,…,n-1, j=1,…,N). Design variables $seg\_angle_{k,j}$ are defined as the difference between the direction (in deg) of the current segment's projection and the projection of the previous one. For the first segment (k=1) of each control polygon $seg\_angle_{1,j}$ is measured with respect to the x-axis (Fig. 2). Additionally, the control points' altitudes $z_{k,j}$ are used as design variables, except for the three fixed points (k=0, k=1, and k=n), which are predefined. For the first segment (k=1), $seg\_length_{1,j}$, and $seg\_angle_{1,j}$ are pre-specified in order to define the initial direction of the path, and they are not included in the design variables of the optimization procedure.

The horizontal coordinates of each B-Spline control point $x_{k,j}$ and $y_{k,j}$ can be easily calculated by using $seg\_length_{k,j}$ and $seg\_angle_{k,j}$ along with the coordinates of the previous control point $x_{k-1,j}$ and $y_{k-1,j}$. The use of $seg\_length_{k,j}$ and $seg\_angle_{k,j}$ as design variables instead of $x_{k,j}$ and $y_{k,j}$ was adopted for three reasons. The first reason is the fact that abrupt turns of each flight path can be easily avoided by explicitly imposing short lower and upper bounds for the $seg\_angle_{k,j}$ design variables. The second reason is that by using the proposed design variables a better convergence rate was achieved compared to the case with the B-Spline control points' coordinates ($x_{k,j}$, $y_{k,j}$, $z_{k,j}$) as design variables. The latter observation is a consequence of the shortening of the search space, using the proposed formulation. The third reason is that by using $seg\_length_{k,j}$ as design variables, an easier determination of the upper bound for each curve's length is achieved, along with a smoother variation of the lengths of each curve's segments. The lower and upper boundaries of each independent design variable are predefined by the user.

For the case of a single vehicle the optimization problem to be solved minimizes a set of five terms, connected to various objectives and constraints; they are associated with the feasibility of the curve, its length and a safety distance from the ground. The cost function to be minimized is defined as:

$$f = \sum_{i=1}^{5} w_i f_i \tag{1}$$

Term f_1 penalizes the non-feasible curves that pass through the solid boundary. In order to compute this term, discrete points along each curve are computed, using B-Spline theory [28] [29] and a pre-specified step for B-Spline parameter u. The value of f_1 is proportional to the number of discrete curve points located inside the solid boundary. Term f_2 is the length of the curve (non-dimensional with the distance between the starting and destination points) and is used to provide shorter paths. Term f_3 is designed to provide flight paths with a safety distance from solid boundaries. For each discrete point i ($i=1,\ldots,nline$, where $nline$ is the number of discrete curve points) of the B-Spline curve its distance from the ground is calculated (the ground is described by a mesh of $nground$ discrete points). Then the minimum distance of the curve and the ground d_{min} is computed. Term f_3 is then defined as:

$$f_3 = \left(d_{safe} / d_{\min} \right)^2, \tag{2}$$

while d_{safe} is a safety distance from the solid boundary.

Term f_4 is designed to provide B-Spline curves with control points inside the pre-specified space. If a control point results with an x or y coordinate outside the pre-specified limits, a penalty is added to term f_4 which is proportional to the violation of the following constraints:

$$\begin{aligned}
&if\ x_{k,j} > x_{\max} \Rightarrow f_4 = f_4 + C_1 * \left| x_{k,j} - x_{\max} \right| \\
&if\ y_{k,j} > y_{\max} \Rightarrow f_4 = f_4 + C_1 * \left| y_{k,j} - y_{\max} \right| \\
&if\ x_{k,j} < x_{\min} \Rightarrow f_4 = f_4 + C_1 * \left| x_{k,j} - x_{\min} \right| \\
&if\ y_{k,j} < y_{\min} \Rightarrow f_4 = f_4 + C_1 * \left| y_{k,j} - y_{\min} \right| \\
&\forall k,\ k = 0,\ldots,n,\ \ \forall j,\ j = 1,\ldots,N,
\end{aligned} \tag{3}$$

where C_1 is a constant, and x_{min}, x_{max}, y_{min}, y_{max} define the borders of the working space. An additional penalty is added to f_4 in case that its value is greater than zero, in order to ensure that curves inside the pre-specified space have a smaller cost function than those having control points outside of it. This can be formally written as

$$if \;\; f_4 > 0 \Rightarrow f_4 = f_4 + C_2 \,, \tag{4}$$

where C_2 is a constant.

Term f_5 was designed to provide path lines within the already scanned terrain. Each control point of the B-Spline curve is checked for whether it is placed over a known territory. The ground is modeled as a mesh of discrete points and the algorithm computes the mesh shell (on the *x-y* plane) that includes each B-Spline control point. If the corresponding mesh shell is characterized as unknown then a constant penalty is added to f_5. A mesh shell is characterized as unknown if all its 4 nodes are unknown (have not been detected by a sensor).

Weights w_i are experimentally determined, using as criterion the almost uniform effect of the last four terms in the objective function. Term $w_1 f_1$ has a dominant role in Eq. 1 providing feasible curves in few generations, since path feasibility is the main concern. The minimization of Eq. 1 results in a set of B-Spline control points, which actually represent the desired path.

For the solution of the minimization problem a Differential Evolution (DE) [30] algorithm is used. The classic DE algorithm evolves a fixed size population, which is randomly initialized. After initializing the population, an iterative process is started and at each generation *G*, a new population is produced until a stopping condition is satisfied. At each generation, each element of the population can be replaced with a new generated one. The new element is a linear combination between a randomly selected element and the difference between two other randomly selected elements. A detailed description of the DE algorithm used in this work can be found in [31].

3 On-Line Path Planning for Cooperating Vehicles

The on-line path planner was designed for navigation and collision avoidance of a small team of autonomous vehicles moving over a completely unknown static 3-D terrain. The general constraint of the problem is the collision avoidance between the vehicles and the ground. The route constraints are: (a) predefined initial and target coordinates for all vehicles, (b) predefined initial directions for all vehicles, (c) predefined minimum and maximum limits of allowed-to-move space. The first two route constraints are explicitly taken into account by the optimization algorithm. The third route constraint is implicitly handled by the algorithm, through the cost function. The cooperation objective is that all members of the team should reach the same target point.

The on-line planner is based on the ideas developed in [16] for a single UAV. The on-line planner rapidly generates a near optimum path, modeled as a 3-D B-Spline curve that will guide each vehicle safely to an intermediate position within the already scanned area. The information about the already scanned area by each vehicle is passed to the rest cooperating vehicles, in order to maximize the knowledge of the environment. The process is repeated until the final position is reached by one or more members of the team (it is possible some members of the team to reach simultaneously the target – in the same number of on-line steps). Then the rest members of the team turn into the off-line mode and a single B-Spline path for each vehicle is computed to guide it from its current position, through the already scanned territory to the common final destination. An alternative approach, which was also tested, is to keep the remaining vehicles in the on-line mode, and not to turn into the off-line mode after a vehicle has reached the target.

In the on-line problem only four control points define each B-Spline curve, the first two of which are fixed and determine the direction of the path of the current vehicle. The remaining two control points are allowed to take any position within the already scanned space, taking into consideration given constraints. The second control is used to make sure that at least first derivative continuity of the two connected curves is provided at their common point. Hence, the second control point of the next curve should lie on the line defined by the last two control points of the previous curve (Fig. 3). The design variables that define each B-Spline segment are the same as in the off-line case, i.e. $seg\_length_{k,j}$, $seg\_angle_{k,j}$, and $z_{k,j}$ (k=2, 3, and j=1,…,N).

The path-planning algorithm considers the scanned surface as a group of quadratic mesh nodes. All ground nodes are initially considered unknown. An algorithm is used to distinguish between nodes visible by the on-board sensors and nodes not visible. The algorithm uses a predefined range R_S for each sensor as well as two angles, one for the horizontal a_H and one for the vertical scanning a_V (Fig. 4). The range and the two angles are predefined by the user and depend on the type of the sensors used. A node is not visible by a sensor if it is not within the sensor's range and angles of sight, or if it is within the sensor's range and angles of sight but is hidden by a ground section that lies between it and the vehicle. The corresponding algorithm, simulates the sensor and checks whether the ground nodes within the sensor's range are "visible" or not and consequently "known" or not. If a newly scanned node is characterized as "visible", it is added to the set of scanned ground nodes, which is common for all cooperating vehicles.

The information from its sensors is used to produce the first path line segment for the corresponding vehicle. As the vehicle is moving along its first segment and until it has traveled about 3/4 of its length, its sensor scans the surrounding area, returning a new set of visible nodes, which are subsequently added to the common set of scanned nodes. This (simulated) scanning is performed for 11 intermediate positions along each path segment. The on-line planner, then, produces a new segment for each vehicle, whose first point is the last point of the previous segment and whose last point lies somewhere in the already scanned area, its position being determined by the on-line procedure. The on-line process is repeated until the ending point of the current path line segment of one vehicle lies close to the final destination. Then the rest members of the team either can turn into the off-line process, in order to reach the target using B-Spline curves that pass through the scanned terrain, or may remain in the on-line mode.

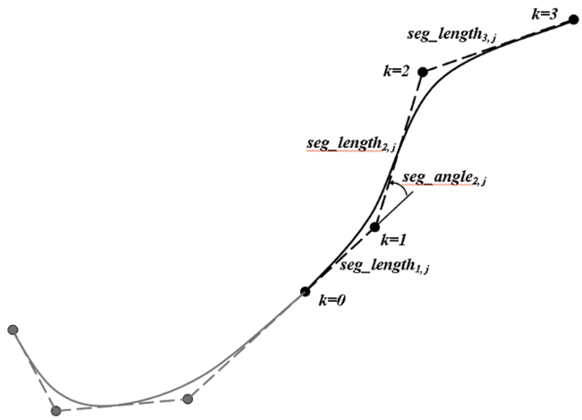

Fig. 3. Schematic representation of the formation of the complete path by successive B-Spline segments (projected on the horizontal plane).

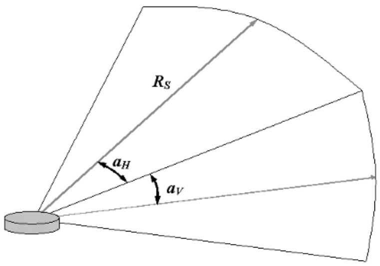

Fig. 4. Schematic representation of the scanned area in front of each vehicle; a_H and a_V are the solid angles in the horizontal and vertical directions that define the scanned sector.

The position at which the algorithm starts to generate the next path line segment for each vehicle (here taken as the 3/4 of the segment length) depends on the range of the sensors, vehicle's velocity and the computational demands of the algorithm. The computation of intermediate path segments for each vehicle is formulated as a minimization problem. The cost function to be minimized is formulated as the weighted sum of seven different terms

$$f = \sum_{i=1}^{7} w_i f_i \ , \tag{5}$$

where w_i are the weights and f_i are the corresponding terms described below.

Terms f_1, f_2, and f_3 are similar to terms f_1, f_3, and f_4 respectively of the off-line procedure. Term f_1 penalizes the non-feasible curves that pass through the solid boundary. Term f_2 is designed to provide flight paths with a safety distance from solid boundaries. Only already scanned ground points are considered for this calculation. Additionally, the points that are lower than a pre-specified (small) vertical distance from the current level of flight are not considered for this calculation. Term f_3 is designed to provide B-Spline curves with control points inside the pre-specified working space.

Term f_4 is designed to provide flight segments with their last control point having a safety distance from solid boundaries. This term was introduced to ensure that the next path segment will not start very close to a solid boundary (which may lead to infeasible paths or paths with abrupt turns). The minimum distance D_{min} from the ground is calculated for the last control point of the current path segment. Only already scanned ground points are considered for this calculation. As in term f_2 the points that are lower than a pre-specified (small) vertical distance from the current level of flight are not considered for this calculation. Term f_4 is then defined as

$$f_4 = \left(d_{safe} / D_{\min}\right)^2 , \tag{6}$$

while d_{safe} is a safety distance from the solid boundary.

The value of term f_5 depends on the potential field strength between the current starting point (of the corresponding path segment) and the final target. This potential field between the two points is the main driving force for the gradual development of each path line in the on-line procedure. The potential is similar to the one between a source and a sink, defined as

$$\Phi = \ln \frac{r_2 + c \cdot r_0}{r_1 + c \cdot r_0} , \tag{7}$$

where r_1 is the distance between the last point of the current curve and the initial point for the current curve segment, r_2 is the distance between the last point of the current curve and the final destination, r_0 is the distance between the initial point of the current curve and the final destination and c is a constant. This potential allows for selecting curved paths that bypass obstacles lying between the starting and ending point of each B-Spline curve [16].

Term f_6 is designed to prevent the vehicles from being trapped in small regions and to force them move towards unexplored areas. Term f_6 repels it from the points of the already computed path lines (of all vehicles). This term has the form

$$f_6 = \frac{1}{N_{po}} \sum_{k=1}^{N_{po}} \frac{1}{r_k} , \tag{8}$$

where N_{po} is the number of the discrete curve points produced so far by all vehicles and r_k is their distance from the last point of the current curve segment.

Term f_7 represents another potential field, which is developed around the final target and has the form

$$f_7 = r_2^2 , \tag{9}$$

where r_2 is the distance between the last point of the current curve and the final destination. Thus, when the vehicle is near its target, the value of this term is quite small and prevents the vehicle from moving away.

Weights w_i in Eq. 5 are experimentally determined, using as criterion the almost uniform effect of all the terms, except the first one. Term $w_1 f_1$ has a dominant role, in order to provide feasible curve segments in a few generations, since path feasibility is the main concern.

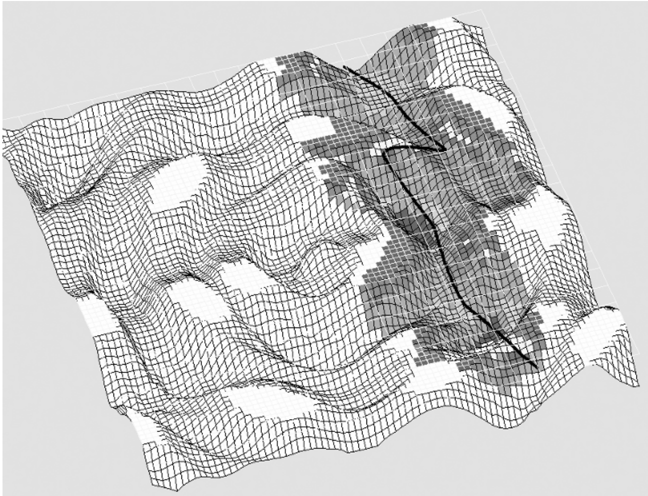

Fig. 5. Test Case 1: On-line path planning for a single UAV. The maximum allowed height for the vehicle is shown using a cutting plane.

4 Simulation Results

The same artificial environment was used for all the test cases considered, with different starting and target points. The artificial environment is constructed within a rectangle of 20x20 (non-dimensional distances). The (non-dimensional) range of the sensors (R_s) that scan the environment was set equal to 4 for all vehicles. The safety distance from the ground was set equal to d_{safe}=0.25. The (experimentally optimized) settings of the Differential Evolution algorithm during the on-line procedure were as follows: *population size* = 20, F = 0.6, C_r = 0.45, *number of generations* = 70. For the on-line procedure we have two free-to-move control points, resulting in 6 design variables. The corresponding settings during the off-line procedure were as follows: *population size* = 30, F = 0.6, C_r = 0.45, *number of generations* = 70. For the off-line procedure eight control points were used to construct each B-Spline curve (including the initial (k=0) and the final one (k=7). These correspond to five free-to-move control points, resulting in 15 design variables. All B-Spline curves have a degree equal to 3.

All experiments have been designed in order to search for path lines between "mountains". For this reason, an upper ceiling has been enforced in the optimization procedure, by explicitly providing an upper boundary for the z coordinates of all B-Spline control points. Test Case 1 corresponds to the on-line path planning for a single vehicle over an unknown environment (Fig. 5). The horizontal and vertical angles a_H and a_V, used for the sensor's simulation, were set equal to 45 degrees. The complete path consists of 6 B-Spline segments; the final curve is smooth enough to be followed by a vehicle. The first turn in the path line is due to the presence of an obstacle (solid ground) in front of the vehicle (Fig. 5); the second turn forces the vehicle towards its final destination.

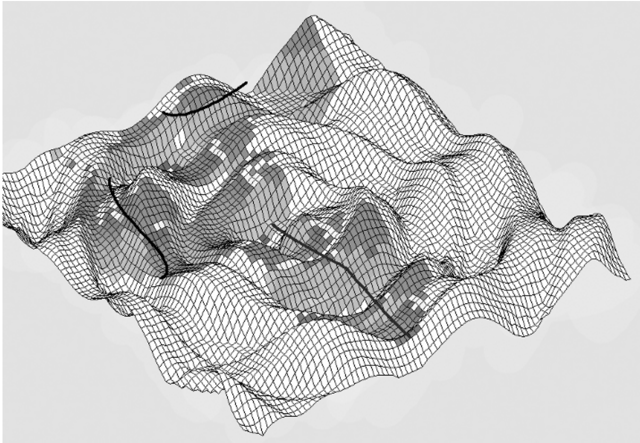

Fig. 6. Test Case 2 corresponds to the on-line path planning for 3 vehicles. The picture shows the status of the path lines when the first vehicle (near the upper corner) reaches the target.

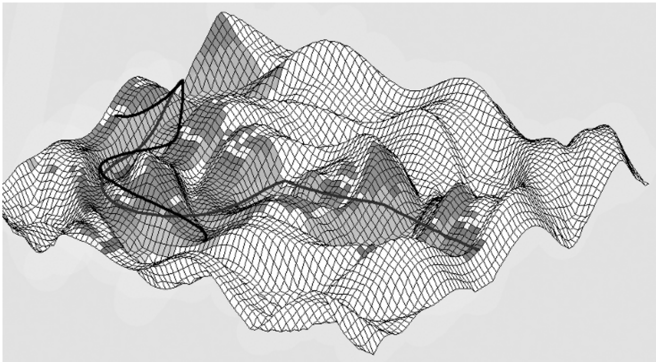

Fig. 7. The final status of the path lines of Test Case 2. The off-line path planner was used by the remaining vehicles to drive them, from their current position to the final destination, through already scanned area.

Test Case 2 corresponds to the on-line path planning for 3 unmanned vehicles (Fig. 1, 6, and 7). The horizontal and vertical angles a_H and a_V, used for the sensor's simulation were set equal to 45 and 30 degrees respectively. Figure 1 shows the status of the three path lines when the first line segment has been computed for all three vehicles. Figure 6 shows the status of the three path lines when the first vehicle reaches the target, after two steps in the on-line procedure. The final status is demonstrated in Fig. 7; the remaining two vehicles turn into off-line mode to reach the target. A curved path is computed for each one of the remaining vehicles, which drives the vehicle from its current position to the target, through the already scanned area.

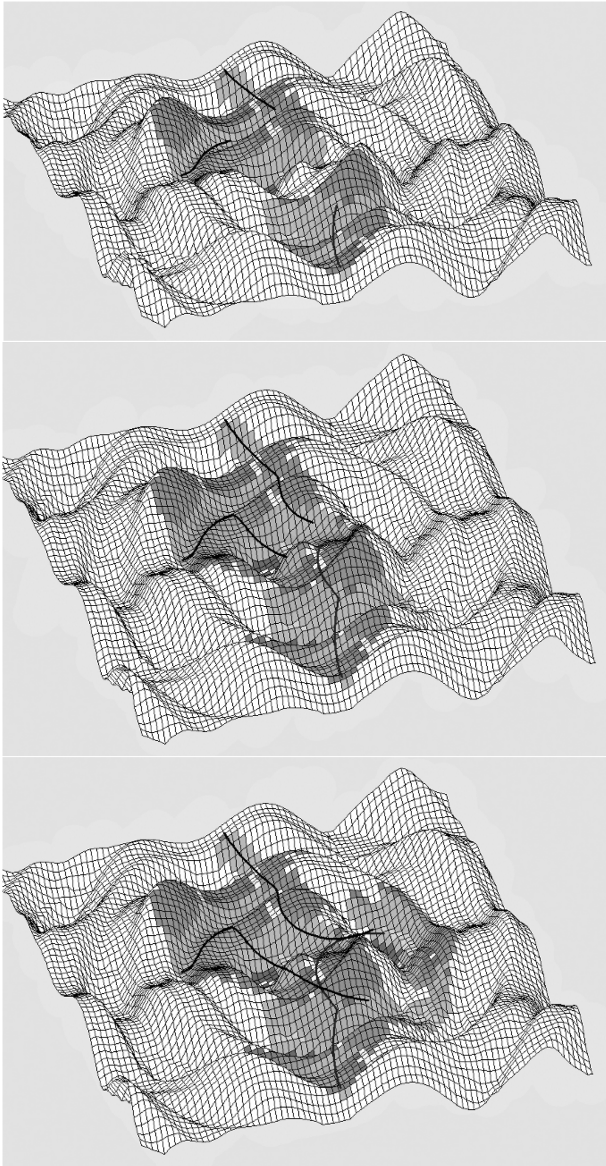

Fig. 8. Test Case 3: Successive snapshots of the path-line for three vehicles, computed using only the on-line planner. Two of the vehicles are reaching the target using 3 segments.

An alternative strategy was considered in Test Case 3. Instead of turning into off-line mode when a vehicle (or more) is reaching the target, the on-line path planner is always used to guide all vehicles to the target. In this case three vehicles are considered. The horizontal and vertical angles a_H and a_V, used for the sensor's simulation were set equal to 45 degrees for both angles. Figure 8 contains successive snapshots of the path lines produced using the on-line path planner. As it can be observed the two vehicles arrive to the target after the same number of steps (Fig. 8). Two more steps of the procedure are needed for the third vehicle to reach the target (Fig. 9).

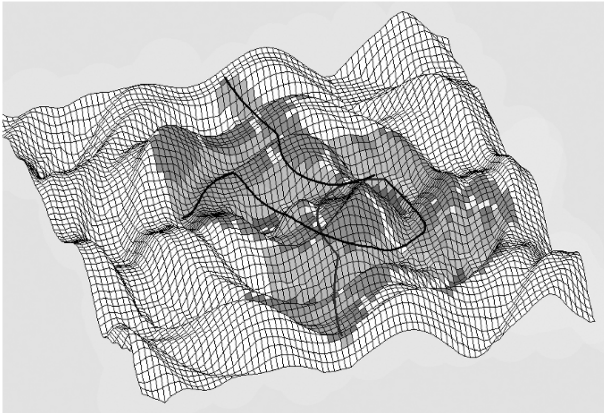

Fig. 9. Test Case 3: Two more steps are needed for the third vehicle to reach the target using the on-line path planner.

5 Discussion

The proposed methodology is applicable to cooperating UAVs but also to cooperating AUVs; in the later case the enforced upper ceiling of the searching space will be the sea surface. Actually, in the case of AUVs the application of the proposed methodology might be easier, due to the lower speed of an AUV compared to an UAV and due to the dynamics of such vehicles. However, in the case of AUVs the suit of the on board sensors will be completely different and the knowledge of the environment will be based on sonar-type sensors. Concerning the application of the proposed methodology to cooperating UAVs, the VTOL type of UAVs seems to be the best choice. The main reason is that the hovering capability of a helicopter may provide the necessary additional time to overcome a computational intensive problem during the calculation of successive curve segments. Additionally, a helicopter has a higher capability to handle abrupt turns, compared to a fixed-wing UAV.

Two issues have to be considered for the application of the proposed methodology to real world scenarios. The first one is the lack of lightweight radar sensors, capable to fit into small UAVs (like small helicopters). Although radar sensors for indoor applications have been already presented (with an effective range of some meters), there is a need for lightweight radar sensors with a range of hundreds of meters, with a weight suitable for small UAVs. The second issue is the communication between the cooperating vehicles. The communication devices should be capable to securely transfer an amount of data (related to the scanned territory by each vehicle), between all cooperating vehicles. Available RF connections for UAV applications are adequate enough for the problem at hand. Acoustic communication links should be used for the communication between AUVs.

References

1. Gilmore, J.F.: Autonomous Vehicle Planning Analysis Methodology. In: Association of Unmanned Vehicles Systems Conference. Washington, DC, pp. 503–509 (1991)
2. LaValle, S.M.: Planning Algorithms. Cambridge University Press (2006)
3. Bortoff, S.: Path Planning for UAVs. In: Amer. Control Conf., Chicago, IL, pp. 364–368 (2000)
4. Szczerba, R.J., Galkowski, P., Glickstein, I.S., and Ternullo, N.: Robust Algorithm for Real-Time Route Planning. IEEE Trans. on Aerosp. Electr. Syst. 36, 869–878 (2000)
5. Zheng, C., Li, L., Xu, F., Sun, F., Ding, M.: Evolutionary Route Planner for Unmanned Air Vehicles. IEEE Trans. on Rob. 21, 609–620 (2005)
6. Uny Cao, Y., Fukunaga, A.S., Kahng, A.B.: Cooperative Mobile Robotics: Antecedents and Directions. Autonomous Robots, 4, 7–27(1997)
7. Schumacher, C.: Ground Moving Target Engagement by Cooperative UAVs. In: 2005 American Control Conference, June 8-10, Portland, OR, USA (2005)
8. Mettler, B., Schouwenaars, T., How, J., Paunicka, J., and Feron E.: Autonomous UAV Guidance Build-up: Flight-Test Demonstration and Evaluation Plan. In: AIAA Guidance, Navigation, and Control Conference, AIAA-2003-5744 (2003)
9. Beard, R.W., McLain, T.W., Goodrich, M.A., Anderson, E.P.: Coordinated Target Assignment and Intercept for Unmanned Air Vehicles. IEEE Trans. on Rob. and Autom. 18 911–922 (2002)
10. Richards, A., Bellingham, J., Tillerson, M., and How., J.: Coordination and Control of UAVs. In: AIAA Guidance, Navigation and Control Conference, Monterey, CA, (2002)
11. Schouwenaars, T., How, J., and Feron, E.: Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA-2004-5141 (2004)
12. Flint, M., Polycarpou, M., and Fernandez-Gaucherand, E.: Cooperative Control for Multiple Autonomous UAV's Searching for Targets. In: 41st IEEE Conference on Decision and Control (2002)
13. Tang, Z., and Ozguner, U.: Motion Planning for Multi-Target Surveillance with Mobile Sensor Agents. IEEE Trans. on Rob. 21, 898–908 (2005)
14. Gomez Ortega, J., and Camacho, E.F.: Mobile Robot Navigation in a Partially Structured Static Environment, using Neural Predictive Control. Control Eng. Practice, 4, 1669–1679 (1996)
15. Kwon, Y.D., and Lee, J.S.: On-Line Evolutionary Optimization of Fuzzy Control System based on Decentralized Population. Intelligent Automation and Soft Computing, 6, 135–146 (2000)
16. Nikolos, I.K., Valavanis, K.P., Tsourveloudis, N.C., Kostaras, A.: Evolutionary Algorithm Based Offline / Online Path Planner for UAV Navigation. IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics, 33, 898-912 (2003)
17. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer (1999)
18. Smierzchalski, R.: Evolutionary Trajectory Planning of Ships in Navigation Traffic Areas. Journal of Marine Science and Technology, 4, 1–6 (1999)
19. Smierzchalski, R., and Michalewicz Z.: Modeling of Ship Trajectory in Collision Situations by an Evolutionary Algorithm. IEEE Trans. on Evol. Comp. 4, 227–241 (2000)
20. Sugihara, K., and Smith, J.: Genetic Algorithms for Adaptive Motion Planning of an Autonomous Mobile Robot. In: 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, California, 138–143 (1997)
21. Sugihara, K., and Yuh, J.: GA-Based Motion Planning for Underwater Robotic Vehicles. UUST-10, Durham, NH (1997)

22. Shima, T., Rasmussen, S.J., Sparks, A.G.: UAV Cooperative Multiple Task Assignments using Genetic Algorithms. In: 2005 American Control Conference, June 8-10, Portland, OR, USA (2005)
23. Moitra, A., Mattheyses, R.M., Hoebel, L.J., Szczerba, R.J., Yamrom, B.: Multivehicle Reconnaissance Route and Sensor Planning. IEEE Trans. on Aerospace and Electronic Syst. 37, 799–812 (2003)
24. Dubins, L.: On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Position. Amer. J. of Math. 79, 497–516 (1957)
25. Shima, T., Schumacher, C.: Assignment of Cooperating UAVs to Simultaneous Tasks Using Genetic Algorithms. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco (2005)
26. Martinez-Alfaro H., and Gomez-Garcia, S.: Mobile Robot Path Planning and Tracking using Simulated Annealing and Fuzzy Logic Control. Expert Systems with Applications, 15, 421–429 (1988)
27. Nikolos, I.K., Tsourveloudis, N., and Valavanis, K.P.: Evolutionary Algorithm Based 3-D Path Planner for UAV Navigation. In: 9th Mediterranean Conference on Control and Automation, Dubrovnik, Croatia (2001)
28. Piegl, L., Tiller, W.: The NURBS Book. Springer (1997)
29. Farin, G.: Curves and Surfaces for Computer Aided Geometric Design, a Practical Guide. Academic Press (1988)
30. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution, a Practical Approach to Global Optimization. Springer-Verlag, Berlin Heidelberg (2005)
31. Nikolos, I.K., Tsourveloudis, N., Valavanis, K.: Evolutionary Algorithm Based Path Planning for Multiple UAV Cooperation. In: Valavanis, K. (ed.), Advances in Unmanned Aerial Vehicles, State of the Art and the Road to Autonomy, pp. 309–340. Springer (2007)