Received January 27, 2018, accepted March 16, 2018, date of publication April 24, 2018, date of current version May 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2824558

Three-Dimensional UAV Routing With Deconfliction

SOHAIL RAZZAQ[1], COSTAS XYDEAS[2], MICHAEL E. EVERETT[3], ANZAR MAHMOOD[4], (Senior Member, IEEE), AND THAMER ALQUTHAMI[5], (Member, IEEE)

[1]Department of Electrical Engineering, COMSATS Institute of Information Technology, Abbottabad 22060, Pakistan
[2]School of Computing and Communications, Lancaster University, Lancaster LA1 4YW, U.K.
[3]Iphestos Systems Ltd., Macclesfield SK11 6QG, U.K.
[4]Department of Electrical (Power) Engineering, Mirpur University of Science and Technology, Mirpur 10250, Pakistan
[5]Department of Electrical and Computer Engineering, King Abdul Aziz University, Jeddah 21589, Saudi Arabia

Corresponding author: Sohail Razzaq (sohailrazzaq@ciit.net.pk)

ABSTRACT The work presented in this paper relates to the conceptual definition, simulation, and performance analysis of a novel, 3-D graph theory-based routing algorithm. The proposed router is designed to provide path planning with deconfliction, i.e., collision avoidance with other moving objects, for unmanned aerial vehicles (UAV) applications in general and autonomous UAV missions in particular. Thus, novel ways for ''sampling'' 3-D operating spaces containing hazard areas and other moving objects are presented. The router operates in a way that conforms to the commonly observed behavior of aircraft flying over relatively long periods of time at constant speed and barometric height. Thus, flying is restricted to a number of possible altitude bands, where changes in altitude and/or direction are restricted to values that ensure excessive airframe stress is avoided. The paper attaches considerable emphasis on algorithmic complexity reduction and develops a novel, adaptive, graph search scheme. The proposed router can also be used to support manned aircraft operation, both for pre-flight planning and during the mission as a pilot aid. Computer simulation results are indicative of a routing system which generates realistic flyable routes while algorithmic complexity is suppressed.

INDEX TERMS 3D UAV routing, adaptive cost function, deconfliction, dynamic graph-theoretic search.

I. INTRODUCTION TO UAV ROUTING

Mission planning in Unmanned Aerial Vehicles (UAVs) requires the identification of a route from base to destination, a route that balances operational factors like flight time, fuel consumption and risks resulting from hazard areas and other flying objects. The work presented in this paper relates to the conceptual definition, simulation and performance analysis of a novel, graph theory based routing algorithm which is designed to provide path planning with deconfliction for UAV applications in general and autonomous UAV missions in particular. The term 'deconfliction' refers to changing the flight path in order to avoid a collision.

At an abstract level there are two possibilities for invoking path planning [1], [2]. First, a route is determined at the control or base station, using appropriate routing algorithm(s). This is often referred to as global or static route planning. Second, on-board route planning is needed during flight time, that is able to take into consideration all the latest task/mission related information, in which case local/dynamic route planning is employed.

Thus global routing i.e. from starting location to destination encompasses solutions using a-priori known mission related information. These route plans are generally of high quality (in an optimization sense) since route plans are produced off-line and there are no severe algorithmic complexity constraints. However during a real world scenario mission, the environment often changes with time and this necessitates ''local routing'' to be performed. Local routing operates on a much smaller scale, as compared to global routing and often requires less information since only part of the whole mission related environment is considered (see for instance [3]). Local planning is also dynamic in nature as information is gathered in real time and the remaining part of the route is optimized accordingly.

Another interesting feature in UAV path planning is Collision Avoidance (CA). Note that the success of

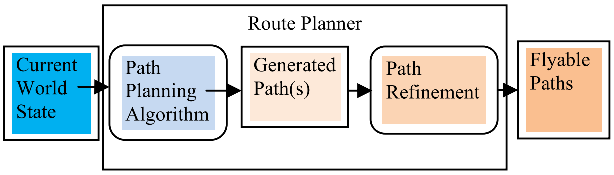


FIGURE 1. A general path planning system design approach.

UAV collision avoidance depends on minimizing UAV response times, since CA is viewed as a "last resort" action. From a global mission planning point of view, collision avoidance is a different "much longer term" issue and one that can be better described as Deconfliction.

Now, the design approach taken in vehicle (including aerial vehicles) path planning often depends on the application under consideration e.g. surveillance, fire fighting, rescue missions. In general however there are some basic building blocks and a possible, simplified, way to represent the process of route planning is given in Figure 1.

"Current world state" (CWS) includes the current position of the vehicle, goal (destination) location and all the static and moving objects and hazard areas in the relevant environment. Given CWS, path planning algorithms generate path(s) which are then refined according to UAV characteristics and nature of application. For example, in the case of aerial vehicles, the derived path must comply with the kinematic characteristics and constraints of the specific vehicle. Note that in some cases path refinement is effectively an integral part of the path planning algorithm.

The remaining sections of this paper are organized as follows. A review of route planning methodologies is provided in section II. The proposed novel 3D Routing algorithm which ensures deconfliction with moving objects is presented in section III. Emphasis is placed on its graph theoretic implementation intricacies, which are specified in a way that favours fast solutions for real time routing in dynamic environments. Section IV deals with the Heuristic component of the cost function used in the path optimization search. Section V presents experimental results and examines system complexity with respect to i) system parameters and ii) the complexity of the given operating environment (world state scenario). Furthermore, complexity reduction and thus fast operation is the main driver behind the introduction and application of a novel adaptive heuristic cost scheme that is discussed in section V. Conclusions are presented in section VI.

II. REVIEW OF ROUTE PLANNING METHODS

Numerous route planning techniques have been proposed in the literature and these can be classified according to their Discrete [4]–[7] or Continuous representation of the operating environment (space) [8]–[10], see Figure 2. In turn this representation often influences the Discrete or Continuous nature of the path finding (optimization) methodology used to define one or several candidate routes (see for instance [11]–[13]). Furthermore, resulting routes must be traceable by the specific UAV under consideration and therefore a route post-processing scheme often follows that produces a "realistic" and "traceable by the UAV" route.

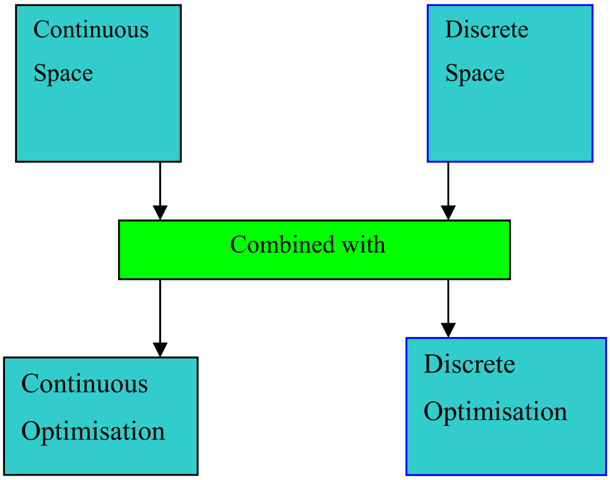


FIGURE 2. Route planning techniques in terms of operating space and optimization methodology characteristics.

Initially path/route planning research was conducted in the field of ground or two dimensional (2D) robotic systems, see [14], [15] and many of the routing techniques currently used in various UAV applications are based on such 2D related work. Vian and Moore [8] and Hebert [16] applied these robotic path planning concepts to aerial vehicles. Path planning for under water autonomous vehicles is presented in [17]–[19]. Furthermore, ground based robotic path planning concepts also have been applied to space applications; see for example [20], [21].

Commencing with Discrete space representations, a 2- or 3-Dimensional environment can be modelled by uniformly sampled 2- or 3-Dimensional grids. For instance, in [22], a world modelling technique is adopted in which the environment is represented by a 3D cubic grid. This technique can be used for route planning and although in this case only a static environment is considered, the method can be modified to operate on applications involving dynamic environments. This 3D cubic grid technique is based on earlier 2D grid models given in [23] and has been applied in robotics related route generation, see [24], [25]. In these techniques, grid nodes are classified as "free" or "occupied" and a Discrete route that consists of free nodes is then defined according to a) simple and predefined node to node motion rules and b) a search process that yields a minimum cost route solution.

In general and for Discrete or Continuous environment type of routers, motion rules may relate to one or a combination of the following cases:

(i) UAV kinematic characteristics,

(ii) simple rules relating to allowable directions of motion, provided that route deconfliction with moving/static objects or hazard areas is observed and

(iii) simple rules based on the distance from objects and hazards.

Type (i) UAV motion specific characteristics are employed usually in continuous environment type routers whose design is based on control theory [26]–[28]. Continuous optimization methods seek an optimal route by defining one or more continuous path curves using a system of nonlinear equations, see [16], [29]. Here, the application of continuous path finding models is considered under multiple optimization constraints and the routing problem is reduced to one of solving path integrals.

Note that Discrete environment routing schemes often employ relatively simple sequential, feed-forward route formulation techniques which cannot backtrack and thus are unable to reconsider other route options. This limitation, although key for achieving low algorithmic complexity, gives rise to suboptimal route solutions. Path formulation schemes which allow backtracking and generate a plethora of possible routes are often combined with efficient route optimisation/selection algorithms. Typical here are graph theoretic based routers and associated optimization techniques [30] with the A$*$ (A star) search algorithm [31], [32] being optimal and many other sub-optimal methods [33]–[35] offering computationally efficient path solutions.

Naturally, alternatives to graph theoretic optimization techniques can be used and evolutionary optimization algorithms have been employed in path planning and collision avoidance in [36]–[38]. The use of evolutionary algorithms for path planning can also be found in [39]–[42]. Another example is integer linear programming [43]. Mixed integer programming is also implemented for path optimization in [44]–[47].

Furthermore motion rules adopted for route formulation may include a degree of randomness [48]–[50]. Following this concept, a random tree search approach is used in [51], [52] where new tree branches are selected randomly. A Rapidly-exploring Random Tree (RRT) based routing algorithm has been applied to UAV path planning in [53]. Positional uncertainty as related to cells of hazardous areas is dealt in [54]. Uncertainty related to cells being free or occupied is taken into consideration in [22] and [23].

Note that the novel routing work presented in this paper is Discrete in terms of both a) its modelling of operating space and associated type (ii) of motion rules and b) its route finding optimization process.

A typical example of type (iii) motion characteristics, which can be used in conjunction with Continuous or Discrete environments, employs a ''potential field'' representation of the operating environment. At each point the field potential is the sum of an attractive and repulsive force with the attractive force being proportional to distance from destination and the repulsive force being inversely proportional to the nearest obstacle or hazard area [55]. The method may not work properly if arc shaped obstacles are present in the environment and modifications have been introduced in order to enhance performance and reduce computational cost, see [56]. A variation of this scheme is presented in [57] in conjunction with genetic search algorithms. Also a potential field based algorithm is proposed in [58] and used to control a multiple UAV mission and path planning.

Returning back to Discrete space type of routers, the UAV operating space can be divided uniformly or non-uniformly into cells which in turn are classified as free or occupied. Cells including part(s) of obstacles or hazard areas are marked as occupied whereas remaining cells are marked as free. Possible routes then pass through adjacent free cells. Thus a search (optimization) algorithm can be applied that examines possible routes and determines a minimum cost route which passes through adjacent free cells. When cells are made strictly non-overlapping, the method is called Exact Cell Decomposition. There are also examples of routers where the non-overlapping condition is relaxed, see [59], [60], and where the Discrete environment representation is the Octree scheme. 2D Quadtrees [61] and Octrees [62] can be used to produce distance based cost maps and corresponding minimum cost path(s). This approach is different from the previously mentioned cubic grid methods [22] in that a hierarchical structure is used to represent the environment effectively at multiple resolutions. The technique is relatively computationally efficient as it allows the route to be calculated relatively quickly over large and ''free from objects'' segments of the environment. Note that the environment under consideration is assumed to be static in [61] and [62] but the general Octree/Quadtree approach can be extended in principle to operate in dynamic environments [63].

In addition to the above techniques, alternative route planning schemes based on Discrete, non-uniformly modelled environment (problem) spaces have been proposed. The Road Map methods of modelling the environment are typical of such schemes. They are based on a network of straight lines connecting the ''start'' and ''destination'' (or goal) locations, with lines not intersecting obstacles and hazard areas. Examples include ''Visibility'' graphs and ''Voronoi'' diagrams.

Visibility graphs [64] generate a line of sight route through a given environment. The route is formed by a connectivity graph network. Obstacles and hazard areas are represented by vertices and only visible vertices, in the sense that each vertex can be seen from the other, are included. A graph search optimization algorithm is often employed to find an appropriate route from start to goal (destination) passing through vertices. Note that such methods may not be appropriate in safety critical applications since the route includes the vertices of ''high risk'' hazardous areas.

Voronoi diagrams [65]–[69] compute a network of line paths within the specified environment with straight line segments in 2D (or planes in 3D) located at the middle distance between hazard areas. Given this network of interconnecting segments, an appropriate and often graph based path search algorithm follows which yields a minimum cost route. However Voronoi diagram based schemes are in principle incapable of trading-off hazard related risks with respect to route length. Note that most of the reported work using Voronoi diagrams is based on 2D diagrams and corresponding route solutions.

Finally trajectory based path planning methods have been proposed in [70] that involve spatial Dubins paths. Shanmugavel used Pythagorean hodographs and 2D clothoids for trajectory based planning [71]. Also, simple trajectories have been used in robotic applications in [72] and [73] and the authors in [74]–[76] used Dubins paths in aerial centered applications. Parametrically described trajectory curves have been employed in path planning and within a discrete optimization framework, in [41] and [77].

III. THREE-DIMENSIONAL ROUTING ALGORITHM WITH DECONFLICTION (3D-RAD)

For an unmanned air vehicle (UAV) to operate in a dynamic environment safely and autonomously of a ground controller (whether by design or circumstance) the aircraft will require an ability to operate within a sense-plan-act type of loop framework, in order to adapt and modify its route plan in a three-dimensional space independently of any ground station support or supervision.

The plethora of path planning techniques presented in section II is indicative of considerable variability in terms of applications and also ensuing limitations, which often characterize such techniques, and thus of the need for improvements. Continuous models which use mainly variational calculus are complex and time consuming. Discrete solutions have the potential to offer high quality, minimum cost routes, but as the size of the problem (search space) grows, they can become computationally very expensive. An approach is therefore required that offers viable trade-offs between computational complexity and ''acceptable for the application domain'' route plans on one hand and a deconfliction capability on the other. Deconfliction here relates to a relatively long term route planning capability which in addition to negotiating static hazard areas, can also deal with moving objects. A novel 3D-RAD which is underpinned by informed graph theoretic search concepts is therefore presented in this section.

A. GENERAL INFORMED GRAPH THEORETIC APPROACH

The routing problem can be stated in geographical notion as: to produce a route between two points in space defined by their latitude, longitude and altitude, while using a ''field'' of values that defines the ''cost'' of moving between any two points. The required route is the one that satisfies some set of pre-defined constraints whilst at the same time minimizing the total cost of traversing the route. Thus the resulting three-dimensional route is defined by a set of points, again represented by latitude, longitude and altitude, which define the required route as a series of segments or ''legs''. Note that more dimensions may be added to the time- space components representing a routing solution e.g. tolerance windows with respect to distance and time of arrival to route points and destination nodes.

In general, the first step when applying a graph theoretic search technique to a continuous search space problem is to sample the space in order to produce a set of discrete sample points; this sampled space is then searched to find a route between the start and goal points that meets mission requirements. Sample points (or nodes) Nj are thus defined as N_jlat, lon, alt where:

- *lat* is the latitude of the sample point (North positive)
- *lon* is the sample point longitude (East positive)
- *alt* is the altitude of the sample point

Pairs of nodes (points) in this sampled search space can be joined by edges; a list of edges (i.e. node pairs) defines node connectivity of a possible route and therefore the following information is required by the search process:

- a list of nodes $N(lat_i, lon_i, alt_i)$
- a list of edge connections between possible route nodes,
- a cost function; the way the cost function is calculated allows for route deconfliction between the UAV and moving hazards
- a start node and
- a goal (i.e. destination) node

The proposed system employs an A∗ type Graph-Theoretic search technique that effectively determines an ordered node list i.e. a route from the start to goal node. A∗ search is an informed best-first search algorithm that reduces the number of graph nodes to be explored by the search process, using an appropriate application specific cost function f(N).

The value of the cost function at each node N_j is:

$$f\left(N_j\right) = g\left(N_j\right) + h\left(N_j\right) \tag{1}$$

where $g(N_j)$ is the actual cost attached to a path $N1, N2, \ldots Nj$ connecting the start node N_1 with node N_j and $h(N_j)$ is a cost estimate of getting from node N_j to the goal node.

Providing this estimate is a lower bound on the actual cost, then the route produced is guaranteed to have the lowest cost function value of all possible routes and is, in this sense, optimal. This estimate of the cost between the current node and the goal (the heuristic) is thus used to focus the search.

Very briefly, the A∗ algorithm [31] starts by examining each ''valid'' node directly connected to the start node. For each of these nodes (its child nodes – the successor set) it calculates the cost {g(N)} of moving to that node and the heuristic value for that node {h(N)}. This set of nodes is retained (and called the OPEN set). The node in the OPEN set with the lowest value of f(N) is then selected (the new parent node), removed from the OPEN set and expanded (i.e. its child nodes identified) to produce its successor set which is then added to the OPEN set. The best node in the OPEN set (i.e. the one with the lowest value of f( ) is again selected, removed from the OPEN set and expanded. The process repeats until either a) the goal node or b) a node sufficiently close to the goal node is reached.[1]

Note that each child node maintains a record of those zone labels that have caused the chosen route to be different to a direct path from parent to goal; these records are then used to annotate the final route nodes with the reasons for this particular node choice. Also note that node expansion is applied within the UAVs operational space which often contains "zones" of hazard areas and moving objects to be avoided, the later type is included due to the deconfliction requirement. Most of the underlying calculations are made using NED coordinate system after applying geodetic to NED transformation. The resulting route nodes are transformed back into geodetic form.

The definition of the cost field, how nodes are expanded and validated and the way the heuristic is calculated are dealt with in the following sub-sections.

B. ALGORITHMIC DETAILS

1) EXCLUSION ZONES

A node is defined to be valid if it does not lie within a "zone" and the path leading to it from its parent node does not cross any zone boundary. Only valid nodes are allowed to be part of a route; the special cases of the start and goal nodes are dealt with separately, see section III.B.4.

A zone is defined by a set of points {lat, lon, alt} lying within the boundaries of one (or more) of the following 3D objects:

Cylinder: is defined by a center point, a radius and upper and lower altitude bounds.

Moving Object safety (exclusion) area: is defined by the object position, a radius (the horizontal safety distance) and a height interval defined by the vertical safety distance.

Polygon: is defined by the positions of the set of vertices and the upper and lower altitude bounds.

Corridor: is defined by the positions of the corridor endpoints, its width and the upper and lower altitude bounds. Corridors are converted to polygons with four vertices.

2) COST FIELD

A "cost field" is defined using cost function values f(N); for valid nodes and throughout experimentation in this paper, cost field values are simply calculated as:

$$f\left(N_j\right) = \sum_{i=1}^{j-1} D\left(N_i, N_{i+1}\right) + h\left(N_j\right) \tag{2}$$

where D(Ni, Ni+1) is the Euclidean distance between the two node points. For invalid nodes the cost field value is infinite.

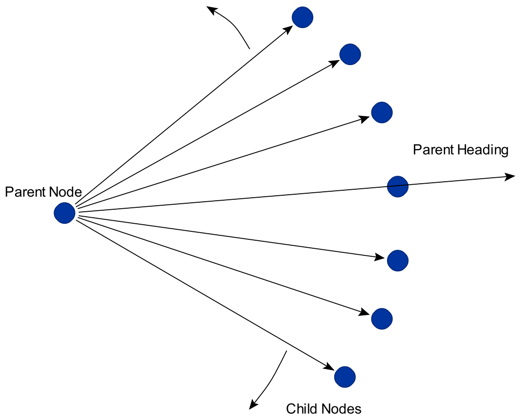


FIGURE 3. **Arc type of node expansion.**

In order to reflect better application requirements and thus improve route quality, additional components can be taken into consideration. These usually take account of risk, due to the presence of hazard zones and other application specific issues, see [68], and can be incorporated into the cost function.

3) NODE EXPANSION

Expanding the node structure on a rectangular grid (the usual case) means that the position of moving objects must be updated for every child node; by expanding in an arc around the parent, this update of positional information is done only once per parent node.

Implementation details of node expansion are given next for a two-dimensional space. The inclusion of the third dimension can then be presented conveniently and in a way that relates to flying aircraft.

a: EXPANSION ARC IN TWO-DIMENSIONAL SPACE

There are two options when expanding a node and create its children; either to expand in the goal direction or in the parent heading. Both cases have their own advantages and disadvantages.

Expanding in the parent heading has two advantages:

Estimated route is smoother,

Computational complexity is reduced, particularly near to objects.

On the other hand, expansion in the goal bearing has the potential to produce shorter routes at the cost of increased complexity. The parent heading option has been selected and used here, mainly due to the complexity issue.

Node expansion takes place by calculating (child) node positions at a range of angles with respect to the parent heading, see Figure 3. Furthermore by making the angle variations symmetrical around the parent heading, change in actual UAV route direction can be restricted to values which relate to maximum aircraft turn rates and thus excessive turning airframe stress is avoided.

An example set of angles used in two-dimensional expansion is given in Table 1.

[1] In some implementations of the $A*$ algorithm, a second set (the CLOSED set) is maintained. This CLOSED set keeps a record of all nodes expanded so far; at each expansion checks are carried out to see if the new node has been visited in the past and compares the old value of f( ) with the new value. This check prevents loops from forming during the search. The way that nodes are expanded in the current algorithm produces a "tree" and ensures that nodes will not be revisited; a CLOSED set is therefore not necessary and this results in considerable savings in calculation time.

TABLE 1. Arc type of node expansion.

0	±10	±20	±30	±40	±50
±60	±80	±100	±125	±150	

Note that the inclusion of very large expansion angles doesn't mean that the resulting route can change so abruptly. Instead it allows the graph search process to be more flexible and often more efficient.

Increasing the number of nodes in the expansion and decreasing the time step between the parent and the child node improves in general overall route quality (in terms of smoothness, distance etc.); this can be at the expense, however, of increasing route calculation time i.e. a trade-off exists between route quality and route estimation time. Experimental results and performance assessment for three dimensional routers are provided in later sections; however it can be mentioned at this point that for two-dimensional routers, using a Time Step (TS) value of 60 seconds for the time interval, an aircraft speed of 100m/s, and the expansion angles of Table 1, reasonable performance is obtained with "typical" 100 Km length routes requiring 0.1 to 5 seconds computation time (experiments run on an Intel(R) Core(TM)2 Duo, 2.4 GHz processor based PC). This wide range of estimation times which has been produced using a conventional A$*$ type search, is an undesirable search characteristic and is addressed later in the paper.

The {lat, lon} position of each expanded node is determined using the set of equations employed in the following section that deals with updating the position of moving objects.

Other node information (the NED Cartesian coordinates, range and bearing to the goal node etc.) is calculated in Cartesian coordinates after transformation using a Geodetic to Cartesian transformation routine.

b: UPDATING THE POSITION OF MOVING OBJECTS

As each node is expanded, the position of moving objects must be updated to the time the UAV will reach a node on an expansion arc. Let the time period required by the UAV to travel from parent to child node be τ; the position of moving object will be updated according to this time. Given that the position, speed and heading of an object is known at the time the aircraft is at the start node, this information can be propagated through the node structure so that the object's position is calculated as required. Assuming that the Earth can be approximated with a sphere over the distances of interest and assuming that the moving object is following a Great Circle path with speed ν and heading hdg at the parent node (i.e. the one being expanded), the object is at position A in Figure 4 with coordinates lat, lon. At any position on the expansion arc, τ seconds later, the object will have moved through d radians, where $d = \nu\tau/Re$ and Re is the Earth radius, to position C with new coordinates lat_n, lon_n.

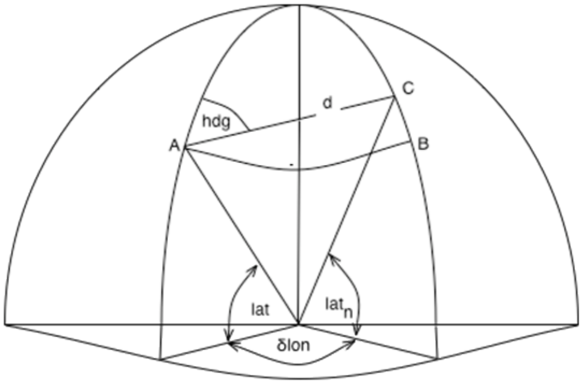


FIGURE 4. Updating moving object position.

Application of the cosine rule for spherical triangles [78], triangle ADC yields the new latitude (lat_n):

$$lat_n = sin^{-1}\left(sin\left(lat\right) cos\left(d\right) cos\left(lat\right) sin\left(d\right) cos\left(hdg\right)\right) \tag{3}$$

Using the same triangle also yields the change in longitude and hence the new longitude:

$$\delta lon = \sin^{-1}\left(\frac{sin\left(hdg\right) sin\left(d\right)}{cos\left(lat\right)}\right) \tag{4}$$

which implies

$$lon_n = lon{-}sin^{-1}\left(\frac{sin\left(hdg\right) sin\left(d\right)}{cos\left(lat\right)}\right) \tag{5}$$

Note that the above equations apply for $d < \pi/2$ and are appropriate to be used here as d is very small compared to $\pi/2$.

c: THREE-DIMENSIONAL NODE EXPANSION

In general, aircraft fly at constant height above mean sea level. This can be approximated as flying at a constant barometric height (given constant air pressure over a region, which in turn equates to a constant distance from the center of the Earth). For this reason, it is better to consider three concentric spheres around the Earth with the middle sphere containing the parent node. Children nodes can be located on the same sphere with the parent, or on one of the other two (i.e. upper and lower) spheres. These two spheres, see Figure 5, are equidistant from the middle one which implies that the UAV can continue flying from node to node without changing its barometric height or that it can change by the same amount of altitude up or down.

Now there are three options available to expand nodes in three dimensions (keeping in mind the necessity of constant barometric height for an aircraft):

(i) Expanding on spheres centered on the parent node keeps the distance travelled constant (and thus the calculation of moving object positions needs to be done only once per parent node). This offers an advantage in terms of calculation time. However, the disadvantage is

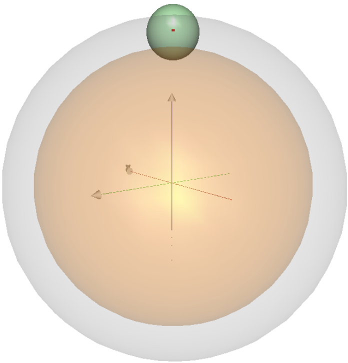


FIGURE 5. Upper (blue) and lower (red) concentric with Earth spheres. Small (green) sphere is centered at parent node.

that the position of nodes within the space will not lie on constant altitude spheres centered on the Earth, so the resulting route will probably require some considerable post-processing to make it flyable.

(ii) An alternative approach would be to have concentric spheres centered at the Earth center and to expand on arcs centered on the parent node and its projection onto the spheres above and below it. The disadvantage of this approach is that the distances to each node are no longer constant, resulting in increased calculation; the advantage is that it keeps sets of nodes at constant altitude which is more natural in terms of aircraft flight and anyway the number of possible altitude bands will generally be restricted.

(iii) A combination of the two above approaches, i.e. finding child nodes that lie on the intersection of the upper, middle and lower spheres (centered on the Earth) with the sphere centered on the parent node is another quite powerful option. In this case one would be able to simultaneously maintain a constant distance from parent to child and keep the altitudes at a set of predetermined levels.

The choice among (i), (ii) and (iii) is down to route calculation time, which in the 2D case and for certain route scenarios can already be too long. Thus, it is necessary to constrain the search space in order to achieve acceptable calculation times and this is easier with either option (ii) or (iii). Option (iii) however gives the best proposition to have constant distance from parent to child as well as discrete altitudes and for this reason, option (iii) has been adopted throughout the experimental work of this paper.

In this case 3D node expansion can be conveniently explained using the three circular plane cuts formed between the upper, middle and lower spheres with the small sphere of radius r centered on the parent node P, see Figure 6. First consider the middle plane circular cut. Here the previously described 2D node expansion is applied with radius r; see Figure 3 and Table 1. Given these nodes NM, expansion nodes at the circumferences of upper or lower circular cuts are defined by shifting NM nodes to a circumference of radius "a" and then shifting them up or down by "h" or "-h" respectively.

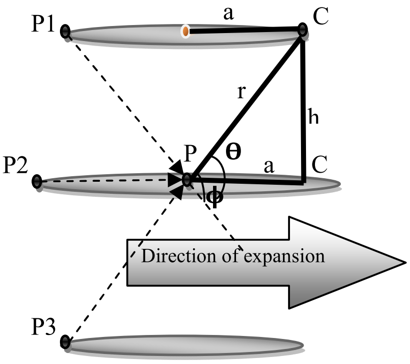


FIGURE 6. 3D node expansion.

TABLE 2. Node expansion angles in new arcs (degrees).

0	+10	+20	-10	-20

Furthermore:

$$h = r\ sin(\theta), \quad a = r\ cos(\theta) \tag{6}$$

The two angles θ and φ in figure 6 are linked via $\varphi = 2\theta$ and relate to two possible node expansion strategies:

(i) The maximum angle of altitude change is θ, which means that if a parent node P was the child of a node in the upper or lower planes (i.e. P1 or P3) then the expansion nodes cannot lie on the same plane with P1 or P3. Alternatively, if P was the child of P2 then expansion nodes can be located in any of the three planes.

(ii) The maximum angle of altitude change is φ which now allows larger than θ altitude changes and thus a parent node P resulting from P1 or P3 can now expand into nodes placed in upper or lower planes respectively.

Note that the number of expansion nodes placed at the upper or lower planes can differ from that used in the middle plane containing the parent node. An example scheme of node expansion angles as applied to upper and lower planes is given in table 2.

4) NODE VALIDATION

As mentioned earlier in section III.B.1, a node is invalid if

i) it lies within an exclusion zone (condition 1) or

ii) the path from the parent node to the node under consideration crosses zone boundaries (condition2)

When a node is declared invalid it is marked as such and is not considered further in the search process.

Also recall that there are two special node cases, i.e. when the node under investigation is either a i) start or ii) goal node.

In the case of a start node the system need only to consider if the node is inside a zone, for all zone types. If the start node is inside a zone then a simple search algorithm finds the

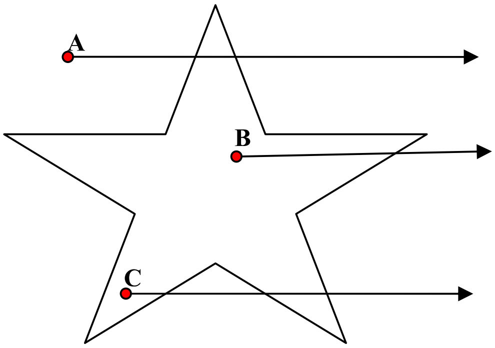


FIGURE 7. Crossings number approach. The direction is that of travel and in this example a horizontal line to the right is employed.

nearest point to the node that is outside the zone and the route is then calculated from this new node.

In the second case, i.e. when a goal node is found to be inside a zone, it is assumed that this zone can be entered and a flag is set so that this zone is ignored in future examinations.

The specific way node validation is performed for different types of hazard area shapes and also for moving objects is discussed next.

a: NODE INSIDE MOVING OBJECT AND CYLINDER TYPE OF ZONES (CONDITION 1)

A node is inside such zone if all of the following are true:

$$\begin{aligned} &Range\ to\ Object's\ centre < Zone\ radius \\ &Node\ altitude < Zone\ upper\ altitude\ limit \\ &Node\ altitude > Zone\ lower\ altitude\ limit \end{aligned}$$

b: NODE INSIDE POLYGON ZONE, INCLUDING AIR CORRIDORS (CONDITION 1)

A crossings number approach is used to determine if the node lies within a polygon zone. Figure 7 illustrates this approach; an infinite half line is projected from the point being examined and the number of times the line crosses the zone boundary are counted. An odd number of crossings indicates that the node is inside the zone, an even (or zero) number that it is outside; hence point A is outside the star (2 crossings), whilst points B (1 crossing) and C (3 crossings) are inside it.

In addition to the above node location tests required with respect to condition 1, the system also needs to ascertain that the path from the parent node to the node under consideration will not infringe upon a zone (i.e. condition 2). Here a node is declared always valid when its altitude is outside the altitude band of the zone. Specific condition 2 test cases for cylinders, moving objects and polygons are described separately below.

c: CYLINDERS (CONDITION 2)

The minimum distance between the cylinder center (in the horizontal plane) and the line segment [A1, A2] defined by the parent and child nodes is calculated; when this distance Ra is less than the cylinder radius the node is invalid.

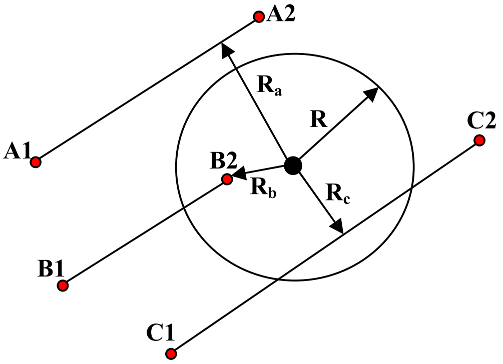


FIGURE 8. Cylinder Zone Infringement.

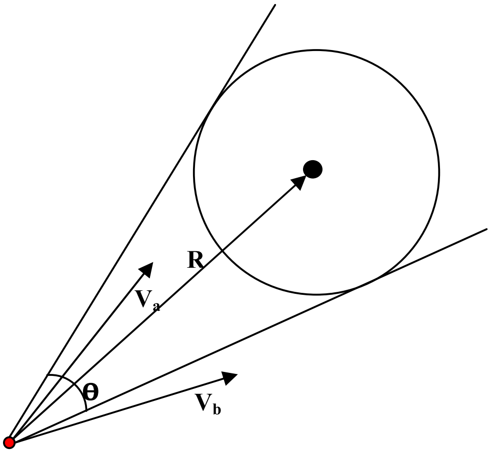


FIGURE 9. Moving object zone avoidance (deconfliction).

Referring to Figure 8 the distance between the line segment A1 to A2 is Ra>R and is clearly "safe"; However Rb, Rc< R, and hence both possible paths B1 to B2 and C1 to C2 are not safe.

d: MOVING OBJECTS (CONDITION 2)

The case of moving objects is not quite so simple since, during the movement of the ownship from parent to child, the object itself will also be moving. In this situation, we consider the relative velocity of the aircraft with respect to the moving object and compare this with the cone angle formed by the range to the object and the diameter of the safety zone, see Figure 9.

If the relative velocity vector is within the cone (e.g. Va) then a collision between the aircraft on this node heading and the object is possible; the time to collision is then calculated and if this time to collision is less than the time step τ the node is declared invalid. When the relative velocity vector lies outside the cone (see Vb) the situation is safe and the node is declared valid.

e: CORRIDORS AND POLYGONS (CONDITION 2)

For polygon zone type, the line segment defined by each vertex pair (V1–V2, V2–V3, V3–V4, V4–V5, V5–V1

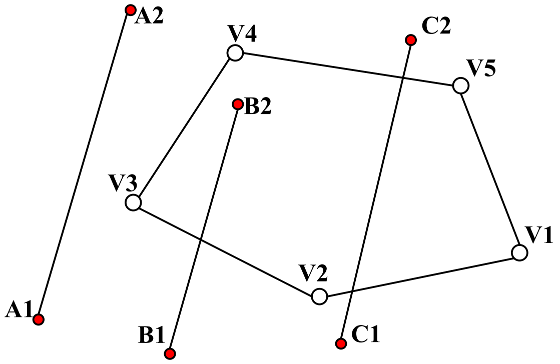


FIGURE 10. Polygon zone avoidance.

in Figure 10) is examined in turn to determine if there is an intersection with the line segment defined by the parent to node path. Any such intersection indicates that the child node is either inside, or on the opposite side of a zone and the node is declared invalid. In Figure 10 therefore, node A2 is valid, whilst B2 and C2 are not.

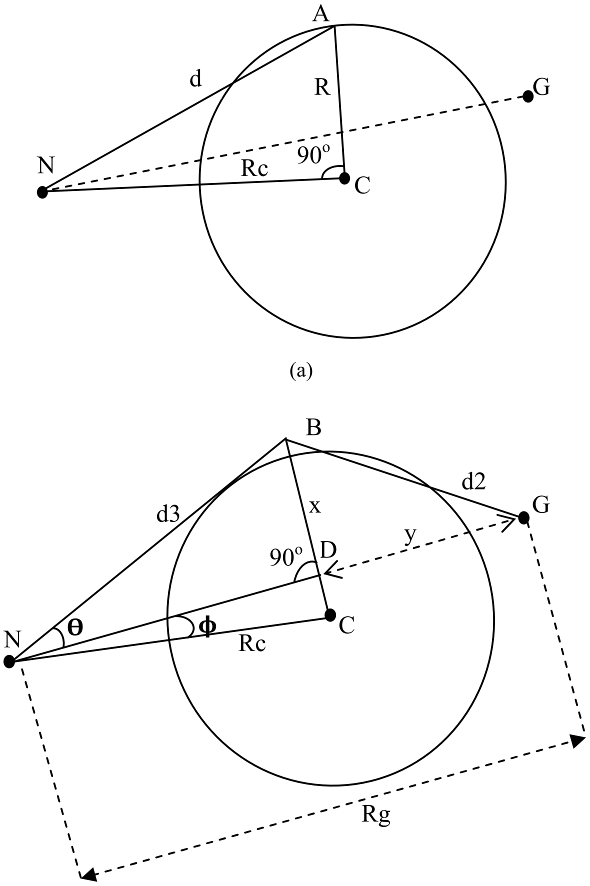
IV. HEURISTIC COST COMPONENT

The heuristic h( ) is one component of the cost function value f( ) = g( ) + h( ) attached to a node and is an estimate of the cost from that node to the goal node. As stated earlier, providing that h( ) is always less than or equal to the actual value of the cost from the node to the goal (i.e. is an under estimate) then the route found will always be optimal in a minimum cost sense. Note that the closer this underestimate cost is to the actual cost, the computational complexity of the node search process tends to decrease. Moreover, significant reduction in route calculation time can be achieved when, at the expense of optimality, this constraint is relaxed. The search algorithm used in this paper is based on true, bounded heuristic values and then applies a weight so that overall complexity is considerably reduced (and cost optimality sacrificed). The cost function thus becomes:

$$f() = g() + \omega h() \tag{7}$$

where ω is the fixed heuristic weight. A value between $1.1 < \omega < 1.5$ for the heuristic weight has been empirically determined to give for the majority of input scenarios "acceptable" routes within reasonable/realistic calculation times. ω is an input system parameter.

Note that search complexity can become prohibitively high when a large static object is in the way of the developing route to the goal. This problem can be avoided by prioritising nodes away from static objects and ω is used to achieve this. Allocating a higher heuristic weight to nodes located near static objects forces the search to show preference to nodes that are relatively far from the objects. This, in turn, reduces search complexity and yields smoother routes. The introduction of ω in the cost function and its effect on algorithmic complexity is further discussed in the Experimental Results section.



(a)

(b)

FIGURE 11. (a) Calculation of cylinder zone heuristic. (b) Calculation of cylinder zone heuristic.

$h_i()$ is calculated with respect to all zones (i) which are intersected by the line "current node to goal" and $h() = MAX\ (h_i())$

A. CYLINDER ZONE RELATED HEURISTIC

$h_i()$ can be defined as the direct distance between the node and the goal, and is given by the range to the goal (i.e. NG in Figure 11a). This distance will always be less than the actual distance and is therefore a heuristic which for cylinder type zones and moving objects preserves optimality in an A$*$ sense.

Of course in order to reduce algorithmic search complexity, the ω factor is introduced in f( ) which may violate the "underestimate" constraint for h( ). Thus an alternative h( ) value has been used that is formed as the sum of two distances, i.e. $h_i() = d_1 + d_2$, see Figures 11a and 11b.

It is assumed here that both current and goal nodes are located on a 2D cut plane of the cylindrical zone.

$$\text{fig 11a, triangle NAC} \quad d_1 = \sqrt{Rc^2 + R^2} \tag{8}$$

$$\begin{aligned} \text{fig. 11b, triangle NBD} \quad x &= d_3 sin\,(\theta - \emptyset) \\ &\approx d_1 \sin\,(\theta - \emptyset) \end{aligned} \tag{9}$$

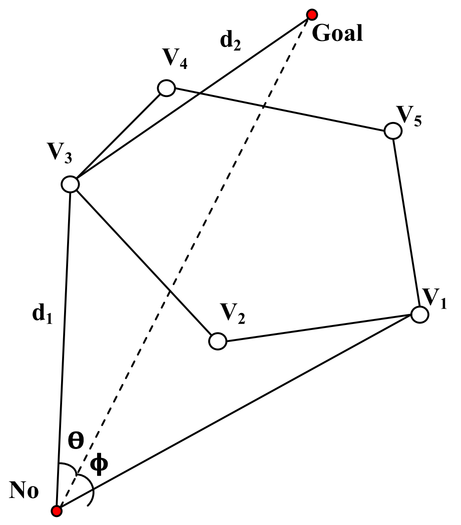


FIGURE 12. Calculation of polygon heuristic.

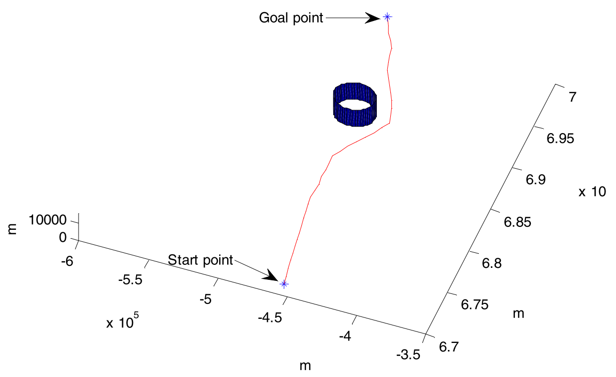


FIGURE 14. Example of 3D-RAD estimated route in a single cylindrical hazard zone (15000 m radius, 0 to 10000 m altitude) input scenario. Fixed $\omega = 1.25$, constant UAV speed 100m/sec, TS = 60secs. Route is defined in terms of 39 segments (legs).

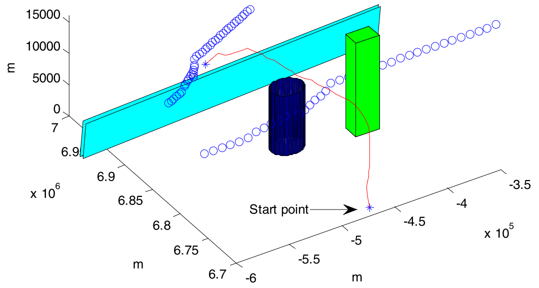


FIGURE 13. A typical scenario with different type of hazards and a generated route.

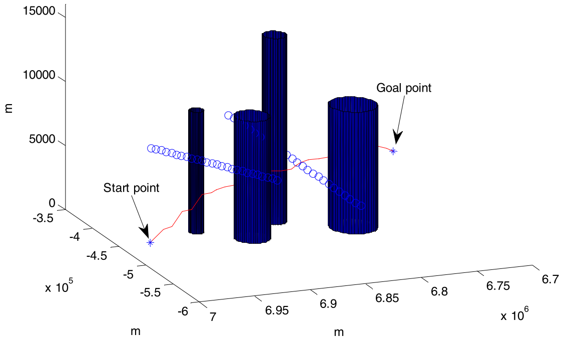


FIGURE 15. 3D-RAD/ ODHW system estimated route. Constant UAV speed 100m/sec, TS = 60secs.

$$\text{fig. 11b,} \quad y = R_g - x\cos(\theta - \emptyset) \tag{10}$$

$$\text{fig. 11b, triangle BGD} \quad d_2 = \sqrt{x^2 + y^2} \tag{11}$$

B. POLYGON & CORRIDOR RELATED HEURISTIC

In the case of polygons and corridors the heuristic is calculated by finding the two vertices that are at the extreme angles from the node to goal path (V1 and V3 in Figure 12) and the one with the smaller angle ($\theta < \emptyset$, hence V3) is chosen. The heuristic is then the distance required to get from the node to the goal via that vertex i.e. $h_i() = d_1 + d_2$.

V. EXPERIMENTAL RESULTS – SEARCH COMPLEXITY ISSUES

Experiments have been carried out in order to investigate system performance as applied to different input scenarios (configurations of UAV operating space). Thus experiments were initially conducted having only one cylindrical zone located somewhere between the start and goal points and these have shown the potential of the system to produce viable route paths, see for example Figures 13 and 14.

However, it became apparent that algorithmic complexity and of course execution time often increased considerably and rapidly to prohibiting levels when the start point is located near to the zone area, or when the route search optimization process considers expanded nodes which are again located too close to the zone. This problem arises as the search algorithm backtracks each time node expansion cannot provide valid or reasonable cost nodes and the search process may iterate (oscillate) back and forth and until a viable path is found. This undesirable behavior and the fact that the heuristic weight ω can be used to bias the search process away from zone areas, (and possible oscillatory search behaviors) led to the development of the following "Object/Zone Dependent Heuristic Weight" scheme.

A. NEW OBJECT DEPENDENT HEURISTIC WEIGHT (ODHW) SCHEME

As mentioned in section IV, allocating a high heuristic weight to nodes located near static objects forces the search process to show preference to nodes that are further away from such exclusion zones. This in turn reduces the algorithm's tendency towards the previously mentioned oscillatory type

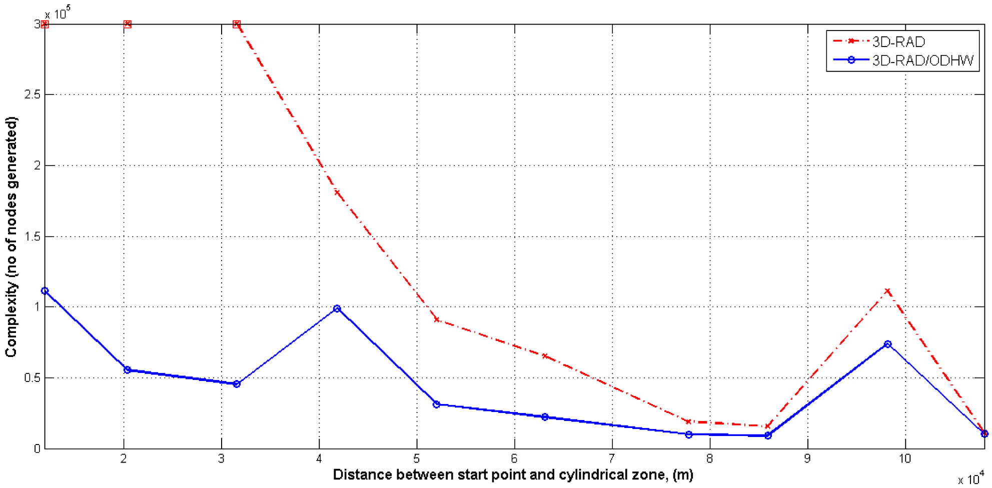


FIGURE 16. 3D-RAD/ODHW versus 3D-RAD complexity characteristics as a function of "start point" to "cylindrical zone" distance in the single zone input scenario of figure 13. Quoted complexity is represented by number of expanded nodes.

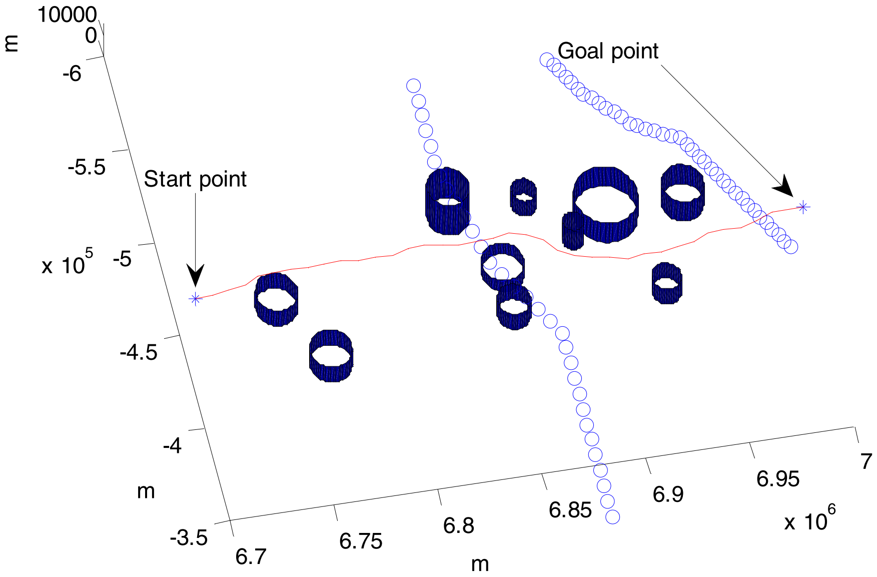


FIGURE 17. 3D-D RAD/ ODHW system estimated route in a 10 hazard areas and two moving objects input scenario.

of search behavior and therefore contains search complexity within acceptable levels. However under normal conditions and when expanded nodes are relatively far away from zones, placing emphasis on the heuristic component of f( ) gives the search a predominately depth-first character, at the expense of breadth-first behavior, and this can prove computationally costly. Thus an ODHW scheme has been developed that switches the value of ω between $\omega 1$ and $\omega 2$ where $1 < \omega 1 < \omega 2$, with i) $\omega 2$ used for "near zone" nodes located within a volume surrounding cylindrical hazard zones uniformly i.e. a hollow type cylinder whose width is X% of the zone radius R, and ii) $\omega 1$ used for all other valid nodes. Note that in the case of non-cylindrical zones a minimum R sphere that completely covers the zone is determined and used in the same way as with cylindrical or spherical zones. The values of $\omega 1$, $\omega 2$ have been defined experimentally as 1.25 and 1.4 respectively. A value of X=10% of the nearest zone's R has been used in experiments. Note that the value of X is also related to input scenario configurations and application (mission) requirements. An example of the type of route that is produced by the proposed 3D-RAD/ ODHW system operating in a space containing several static hazard zones and moving objects is shown in Figure 15.

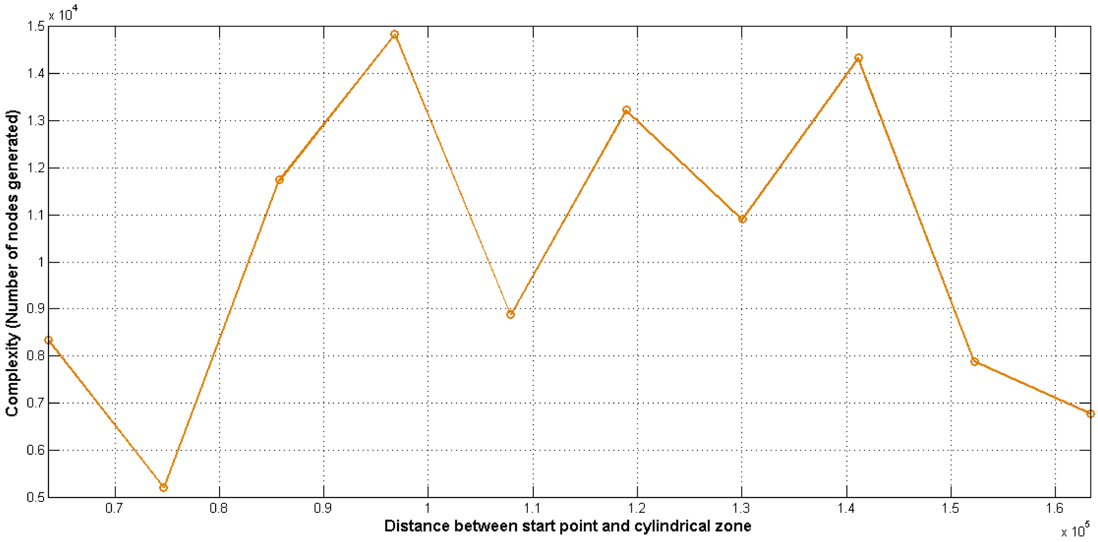


FIGURE 18. 3D-RAD/ODHW complexity characteristics as a function of "start point" to first "cylindrical zone" distance in the multi zone input scenario of figure 15.

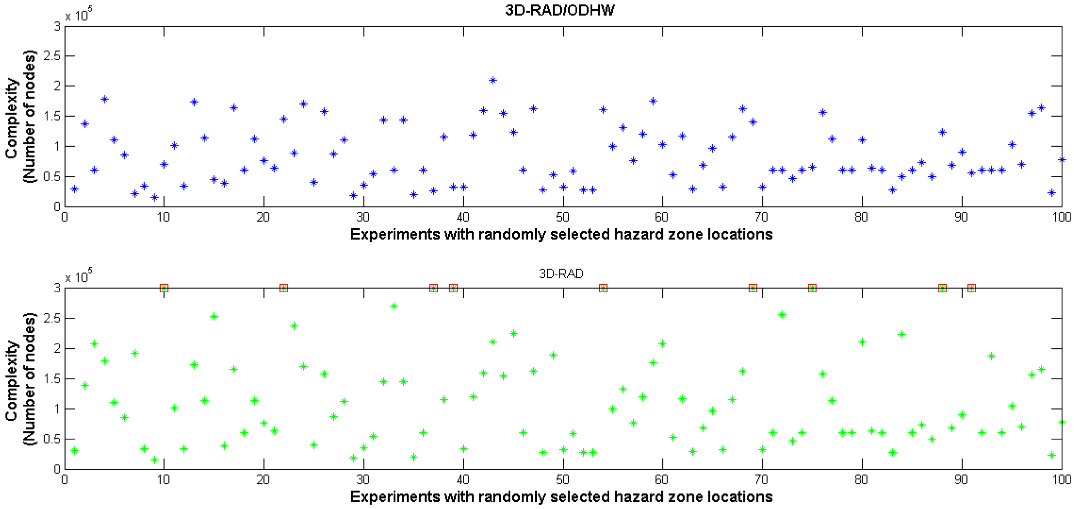


FIGURE 19. (a) top and (b) bottom scatter graphs correspond to 3D-RAD/ODHW and 3D-RAD complexity values measured over 100 different, 10 zones, input scenario experiments.

Figure 16 illustrates the search related computational complexity benefits realized by allowing the heuristic weight to adapt its value according to the proximity of a valid node to an exclusion area i.e. by incorporating ODHW. In this figure search complexity is measured in terms of number of expanded nodes. Here an (example) upper limit (UL) of 300,000 nodes is taken as cut-off value (infinity) and used to terminate the search process on the basis that the time taken to estimate the required route is too long for the system to be of practical use. Of course, the actual value of UL depends on the application domain characteristics and the hardware realization of the router. Furthermore, search complexity in figure 16 is measured as the start point moves closer to the hazard zone of figure 13, for both 3D-RAD (fixed heuristic weight) and 3D-RAD/ODHW (adaptive heuristic weight) systems. Note that the computational complexity of both schemes is almost the same up to a specific "start point" to "hazard zone" distance. When this distance is further reduced 3D-RAD search complexity becomes prohibitively large very rapidly, whereas 3D-RAD/ODHW continues to perform at the same average complexity level. Also note that although Figure 15 experimentation is based on a specific input scenario, is nevertheless indicative of the general search complexity behavior of the two systems.

An example route generated by 3D-RAD/ODHW operating in a multi Hazard zones and moving objects input scenario is shown in Figure 17. Figure 18 illustrates 3D-RAD/ODHW system search complexity as the start point location moves closer to the first zone encountered in the start point to goal point direction. Note that for this multi zone related input

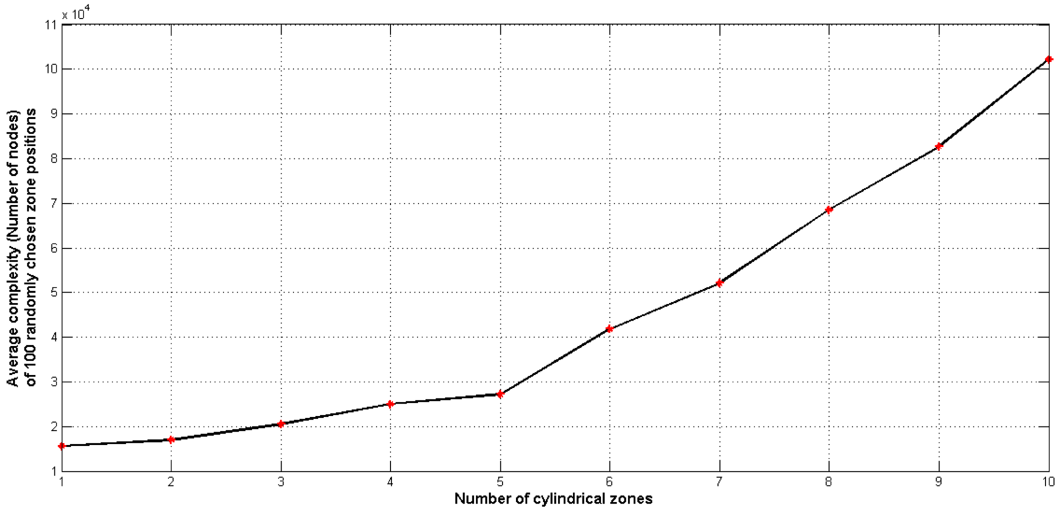


FIGURE 20. 3D-RAD/ ODHW search complexity as a function of number of static hazard zones.

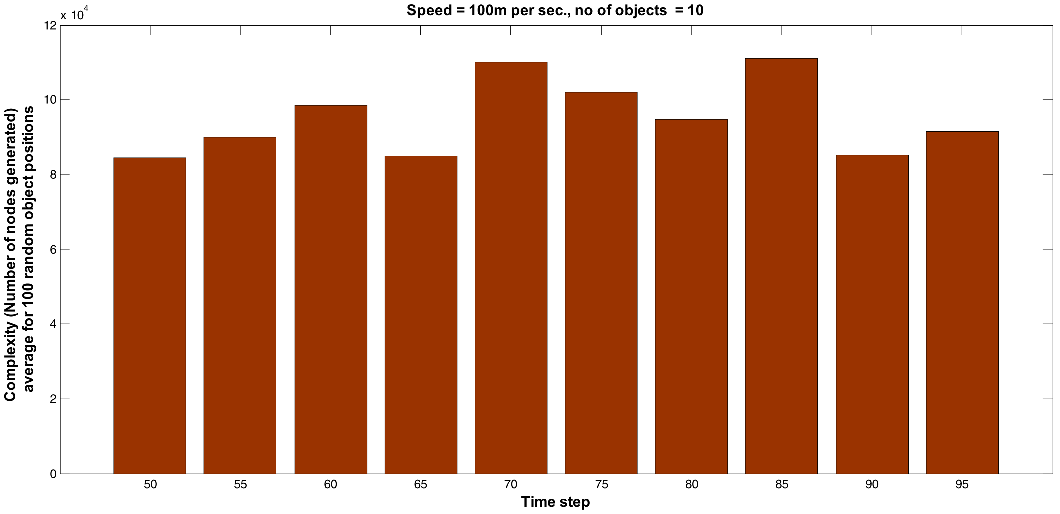


FIGURE 21. Search complexity as a function of expansion step size

scenario, search complexity varies considerably but is bounded and remains well within acceptable levels. A more complete understanding of the variability in search complexity which 3D-RAD/ODHW and 3D-RAD exhibit with respect to the location of 10 hazard zones, is shown in Figures 19a and 19b respectively and over 100 different input scenarios.

Note that 3D-RAD complexity exceeds the 300K set upper limit in nine out of the one hundred randomly generated input scenarios. On the other hand, 3D-RAD/ ODHW complexity is bounded and, in most cases, below 60% of the maximum allowable value.

B. COMPLEXITY AS A FUNCTION OF NUMBER OF HAZARD AREAS AND MOVING OBJECTS

In general search complexity is expected to increase as the number of hazard zones and objects increases. This is shown below in Figure 20 where complexity is experimentally measured as the number of static zones increases from one to ten. Note that figure 19 displays average complexity values derived from 100 experiments per number of static zones,

with zones randomly positioned in each experiment. Furthermore, cylindrical zones are generated having radius and height in the range 5000m to 10000m and 0m to 10000m, respectively. These typical, relatively low complexity experimental results show the general trend of increasing complexity as the number of static objects in the environment increases.

C. COMPLEXITY VERSUS NODE EXPANSION TIME STEP

A first reaction to the question of how search complexity varies with respect to node expansion Time-Step (TS) (given a constant UAV speed) may be that search complexity increases as the time- step size decreases. However, making TS larger, in order to reduce complexity, has an effect on the behavior of the graph theoretic search algorithm itself, which in turn relates to the topology of the input scenario. That is, the relationship between the Distance separating Hazard Zones (DHZ) and TS affects to a large extent algorithmic search complexity.

In general, small TS (and corresponding travelling distance) values, as compared to DHZ, allow the search process to easily find routes passing in between zones with minimum search oscillatory behavior. As TS increases, search space resolution decreases but gains in complexity are compensated by the possible introduction of oscillatory node expansion behavior.

Thus as TS values increase, say from 50 to 95 sec, search complexity is confined within a relatively small range, see Figure 21. In this figure and for each time step TS, measured complexity is averaged over 100 different input scenarios with each scenario containing 10 randomly located hazard zones.

VI. CONCLUSIONS

The capability of defining viable flying routes for UAVs in general and autonomous UAVs in particular, is an integral part of the planning work done prior to a mission and, much more importantly, during the mission when routes must be modified in response to changes in the operating environment.

Among the plethora of possible design methodologies, work in this paper is focused on a graph theoretic (GT) search approach whereby estimated UAV routes are discrete in nature and defined in terms of nodes (i.e. points in space) and associated segments. Having decided to employ a GT representation of the problem space, the paper developed novel ways of ‘‘sampling’’ 3D operating spaces containing hazard areas and other moving objects.

In particular node expansion has been defined and used, within a minimum cost function optimization framework, in a way that conforms to flying behaviors. That is aircraft mostly fly with constant speed, at a constant barometric height and for relatively long periods of time.

Furthermore, flying is often restricted to a number of possible altitude bands whereas changes in altitude and/or direction are restricted to values that ensure excessive airframe stress is avoided.

Of course, the router should be able to provide output(s) within a limited time period, particularly when operating within a rapidly changing mission environment. This makes router algorithmic complexity a major issue.

Thus the paper examines the computational characteristics of the proposed GT methodology and their dependency on the operational environment, i.e. location and number of static or moving hazard zones and objects. Experimentation has revealed that the level of GT search complexity increases considerably when expanded nodes are located near to zone areas. This undesirable search behavior is effectively rectified in a novel way with the introduction of an adaptive weighting factor which operates on the Heuristic part of the cost function.

The resulting 3D router offers flyable routes with deconfliction, i.e. collision avoidance with other moving objects. The assumption made of moving objects flying at constant speed is not restrictive, particularly when autonomous UAVs and hence their onboard routers operate in a Sense-Plan-Act type of planning framework where routes are re-estimated using up-to-date information.

REFERENCES

[1] A. Tsourdos, B. White and M. Shanmugavel, *Cooperative Path Planning of Aerial Vehicles*. Hoboken, NJ, USA: Wiley, 2011.

[2] M. K. M. Kapadia, ‘‘Geometric and discrete path planning for interactive virtual worlds,’’ in *Geometric and Discrete Path Planning for Interactive Virtual Worlds*, vol. 1. San Rafael, CA, USA: Morgan Claypool, 2016, p. 201, doi: 10.2200/S00687ED1V01Y201512VCP023.

[3] N. Wen, L. Zhao, X. Su, and P. Ma, ‘‘UAV online path planning algorithm in a low altitude dangerous environment,’’ *IEEE/CAA J. Autom. Sinica*, vol. 2, no. 2, pp. 173–185, Apr. 2015.

[4] M. Davoodi, F. Panahi, A. Mohades, and S. N. Hashemi, ‘‘Multi-objective path planning in discrete space,’’ *Appl. Soft Comput.*, vol. 13, no. 1, pp. 709–720, 2013. [Online]. Available: https://doi.org/10.1016/j.asoc.2012.07.023

[5] D. H. Lewis, *Optimal Three-Dimensional Path Planning Using Visibility Constraints*. Monterey, CA, USA: Naval Postgraduate School, 1988.

[6] J. Leary, *Search for a Stealthy Flight Path Through a Hostile Radar Defense Network*. Monterey, CA, USA: Naval Postgraduate School, 1995.

[7] J. Kim and J. P. Hespanha, ‘‘Discrete approximations to continuous shortest-path: Application to minimum-risk path planning for groups of UAVs,’’ in *Proc. 42nd IEEE Conf. Decision Control*, Dec. 2003, pp. 1734–1740.

[8] J. L. Vian and J. R. Moore, ‘‘Trajectory optimization with risk minimization for military aircraft,’’ *J. Guid., Control, Dyn.*, vol. 12, no. 3, pp. 311–317, 1989.

[9] M. C. Novy, D. R. Jacques, and M. Pachter, ‘‘Air vehicle optimal trajectories between two radars,’’ in *Proc. Amer. Control Conf.*, May 2002, pp. 785–790.

[10] M. Zabarankin, S. Uryasev, and R. Murphey, ‘‘Aircraft Routing under the Risk of Detection,’’ *Naval Res. Logistics*, vol. 53, no. 8, pp. 728–747, 2006.

[11] P. Yang, K. Tang, J. A. Lozano, and X. Cao, ‘‘Path planning for single unmanned aerial vehicle by separately evolving waypoints,’’ *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1130–1146, Oct. 2015.

[12] X. Sun, Y. Liu, W. Yao, and N. Qi, ‘‘Triple-stage path prediction algorithm for real-time mission planning of multi-UAV,’’ *Electron. Lett.*, vol. 51, no. 19, pp. 1490–1492, Sep. 2015.

[13] Y. Lin and S. Saripalli, ‘‘Sampling-based path planning for UAV collision avoidance,’’ *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3179–3192, Nov. 2017.

[14] J. G. Bender, ‘‘An overview of systems studies of automated highway systems,’’ *IEEE Trans. Veh. Technol.*, vol. 40, no. 1, pp. 82–99, Feb. 1991.

[15] S. E. Shladover *et al.*, ‘‘Automated vehicle control developments in the PATH program,’’ *IEEE Trans. Veh. Technol.*, vol. 40, no. 1, pp. 114–130, Feb. 1991.

[16] J. M. Hebert, *Air Vehicle Path Planning, Graduate School of Engineering and Management*. Wright-Patterson AFB, OH, USA: Air Force Inst. Technol., 2001.

[17] J. Yuh, ‘‘Underwater robotics,’’ in *Proc. IEEE Conf. Robot. Autom.*, Apr. 2000, pp. 932–937.

[18] T. R. Smith, H. Hanßmann, and N. E. Leonard, ‘‘Orientation control of multiple underwater vehicles with symmetry-breaking potentials,’’ in *Proc. 40th IEEE Conf. Decision Control*, Dec. 2001, pp. 4598–4603.

[19] P. Oliveira, A. Pascoal, V. Silva, and C. Silvestre, ‘‘Mission control of the MARIUS AUV: System design, implementation, and sea trials,’’ *Int. J. Syst. Sci.*, vol. 29, pp. 1065–1080, Jun. 1998.

[20] P. Tompkins, A. Stentz, and D. Wettergreen, ‘‘Global path planning for mars rover exploration,’’ in *Proc. IEEE Aerosp. Conf.*, Mar. 2004, pp. 801–815.

[21] D. B. Gennery, ‘‘Traversability analysis and path planning for a planetary rover,’’ *Auto. Robots*, vol. 6, no. 2, pp. 131–146, 1999.

[22] Y. Roth-Tabak and R. Jain, "Building an environment model using depth information," *Computer*, vol. 22, no. 6, pp. 85–90, Jun. 1989.

[23] L. Matthies and A. Elfes, "Integration of sonar and stereo range data using a grid-based representation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1988, pp. 727–733,

[24] S. Bandi and D. Thalmann, "Path finding for human motion in virtual environments," *Comput. Gemoetry*, vol. 15, pp. 103–127, Feb. 2000.

[25] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.

[26] I. Duleba and J. Z. Sasiadek, "Nonholonomic motion planning based on Newton algorithm with energy optimization," *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 3, pp. 355–363, Mar. 2003.

[27] E. Frazzoli, "Maneuver-based motion planning and coordination for multiple UAVs," in *Proc. 21st Digit. Avionics Syst. Conf.*, 2002, p. 8D3.

[28] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *J. Intell. Robot. Syst.*, vol. 57, no. 1, pp. 65–100, 2010.

[29] I. M. Mitchell and S. Sastry, "Continuous path planning with multiple constraints," in *Proc. IEEE Conf. Decision Control*, Dec. 2003, pp. 5502–5507.

[30] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Math.*, vol. 1, no. 1, pp. 269–271, 1959.

[31] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[32] F. H. Tseng, T. T. Liang, C. H. Lee, L. D. Chou, and H. C. Chao, "A star search algorithm for civil UAV path planning with 3G communication," in *Proc. 10th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Kitakyushu, Japan, Aug. 2014, pp. 942–945.

[33] R. E. Korf, "Depth-first iterative-deepening: An optimal admissible tree search," *Artif. Intell.*, vol. 27, no. 1, pp. 97–109, 1985.

[34] Y. Björnsson, M. Enzenberger, R. C. Holte, and J. Schaeffer, "Fringe search: Beating A* at pathfinding on game maps," in *Proc. IEEE Symp. Comput. Intell. Games (CIG)*, Colchester, U.K., Apr. 2005.

[35] S. Russell, "Efficient memory-bounded search methods," in *Proc. 10th Eur. Conf. Artif. Intell.*, Vienna, Austria, 1992, pp. 1–5.

[36] S. Mittal and K. Deb, "Three dimensional offline path planning for UAVs using multiobjective evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sep. 2007, pp. 3195–3202.

[37] D. Zhang, Y. Xian, J. Li, G. Lei, and Y. Chang, "UAV path planning based on chaos ant colony algorithm," in *Proc. Int. Conf. Comput. Sci. Mech. Autom. (CSMA)*, Hangzhou, China, Oct. 2015, pp. 81–85.

[38] J. Li, Y. Huang, Z. Xu, J. Wang, and M. Chen, "Path planning of UAV based on hierarchical genetic algorithm with optimized search region," in *Proc. 13th IEEE Int. Conf. Control Autom. (ICCA)*, Jul. 2017, pp. 1033–1038.

[39] D. Jia and J. Vagners, "Parallel evolutionary algorithms for UAV path planning," in *Proc. AIAA 1st Intell. Syst. Techn. Conf.*, Chicago, IL, USA, 2004, p. 6230.

[40] C. Zheng, L. Li, F. Xu, F. Sun, and M. Ding, "Evolutionary route planner for unmanned air vehicles," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 609–620, Aug. 2005.

[41] I. K. Nikolos, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, "Evolutionary algorithm based offline/online path planner for UAV navigation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 6, pp. 898–912, Dec. 2003.

[42] X. Wu, Z. Feng, J. Zhu, and R. Allen, "GA-based path planning for multiple AUVs," *Int. J. Control*, vol. 80, no. 7, pp. 1180–1185, 2007.

[43] J. Eele and A. Richards, "Rapid updating for path-planning using nonlinear branch-and-bound," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, May 2010, pp. 3575–3580.

[44] T. A. Ademoye, A. Davari, and W. Cao, "Three dimensional obstacle avoidance maneuver planning using mixed integer linear programming," in *Proc. Robot. Appl.*, Honolulu, HI, USA, 2006.

[45] T. Schouwenaars, E. Feron, and J. How, "Multi-vehicle path planning for non-line of sight communication," in *Proc. Amer. Control Conf.*, Minneapolis, MN, USA, 2006, p. 6.

[46] A. Richards, J. How, T. Schouwenaars, and T. Feron, "Plume avoidance maneuver planning using mixed integer linear programming," in *Proc. AIAA Guid., Navigat. Control Conf.*, 2001, pp. 6–9.

[47] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proc. Amer. Control Conf.*, 2002, pp. 1936–1941.

[48] J. N. Eagle and J. R. Yee, "An optimal branch-and-bound procedure for the constrained path, moving target search problem," *Oper. Res.*, vol. 38, no. 1, pp. 110–114, 1990.

[49] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[50] P. O. Pettersson and P. Doherty, "Probabilistic roadmap based path planning for an autonomous unmanned aerial vehicle," in *Proc. 14th Int. Conf. Auto. Planning Scheduling (ICAPS)*, 2004, pp. 1–7.

[51] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Impact of workspace decompositions on discrete search leading continuous exploration (DSLX) motion planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2008, pp. 3751–3756.

[52] T. Ju, S. Liu, J. Yang, and D. Sun, "Apply RRT-based path planning to robotic manipulation of biological cells with optical tweezer," in *Proc. Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2011, pp. 221–226.

[53] M. Kothari, I. Postlethwaite, and D.-W. Gu, "Multi-UAV path planning in obstacle rich environments using rapidly-exploring random trees," in *Proc. 48th IEEE Conf. Decision Control*, Dec. 2009, pp. 3069–3074.

[54] M. Jun and R. D'Andrea, "Path planning for unmanned aerial vehicles in uncertain and adversarial environments," in *Cooperative Control: Models, Applications and Algorithms*. Norwell, MA, USA: Kluwer, 2003, pp. 95–110.

[55] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Mar. 1985, pp. 500–505.

[56] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.

[57] F. A. Cosío and M. A. P. Castañeda, "Autonomous robot navigation using adaptive potential fields," *Math. Comput. Model.*, vol. 40, pp. 1141–1156, Nov. 2004.

[58] Y. Eun and H. Bang, "Cooperative control of multiple unmanned aerial vehicles using the potential field theory," *J. Aircraft*, vol. 43, no. 6, pp. 1805–1814, 2006.

[59] S. Kambhampati and L. Davis, "Multiresolution path planning for mobile robots," *IEEE J. Robot. Autom.*, vol. 2, no. 3, pp. 135–145, Sep. 1986.

[60] D. Jung and K. K. Gupta, "Octree-based hierarchical distance maps for collision detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1996, pp. 454–459.

[61] J. Vörös, "A strategy for repetitive neighbor finding in images represented by quadtrees," *Pattern Recognit. Lett.*, vol. 18, no. 10, pp. 955–962, 1997.

[62] J. Vörös, "A strategy for repetitive neighbor finding in octree representations," *Image Vis. Comput.*, vol. 18, no. 14, pp. 1085–1091, 2000.

[63] Y. Kitamura, T. Tanaka, F. Kishino, and M. Yachida, "3-D path planning in a dynamic environment using an octree and an artificial potential field," in *Proc. Int. Conf. Intell. Robots Syst.*, Pittsburgh, PA, USA, 1995, pp. 474–481.

[64] M. D. Berg, O. Cheong, M. V. Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed. Berlin, Germany: Springer-Verlag, 2008.

[65] S.-M. Li *et al.*, "Autonomous hierarchical control of multiple unmanned combat air vehicles (UCAVs)," in *Proc. Amer. Control Conf.*, Anchorage, AK, USA, 2002, pp. 274–279.

[66] R. W. Beard, T. W. McLain, and M. A. Goodrich, "Coordinated target assignment and intercept for unmanned air vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, USA, May 2002, pp. 2581–2586.

[67] S. Garrido, L. Moreno, D. Blanco, and P. Jurewicz, "Path planning for mobile robot navigation using voronoi diagram and fast marching," *Int. J. Robot. Autom.*, vol. 2, no. 1, pp. 42–64, 2011.

[68] P. Bhattacharya and M. Gavrilova, "Roadmap-based path planning—Using the voronoi diagram for a clearance-based shortest path," *IEEE Robot. Autom. Mag.*, vol. 15, no. 2, pp. 58–66, Jun. 2008.

[69] S. Bortoff, "Path planning for unmanned aerial vehicles," in *Proc. Amer. Control Conf.*, Chicago, IL, USA, 2000, pp. 364–368.

[70] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, no. 3, pp. 497–516, 1957.

[71] M. Shanmugavel, A. Tsourdos, R. Zbikowsky, B. A. White, C. A. Rabbath, and N. Lechevin, "A solution to simultaneous arrival of multiple UAVs using Pythagorean Hodograph curves," in *Proc. Amer. Control Conf.*, Minneapolis, MN, USA, 2006, p. 6.

[72] D. Zhang, L. Wang, and J. Yu, "Geometric topology based cooperation for multiple robots in adversarial environments," *Control Eng. Pract.*, vol. 16, no. 9, pp. 1092–1100, 2008.

[73] T. Shima, S. J. Rasmussen, and A. G. Sparks, "UAV cooperative multiple task assignments using genetic algorithms," in *Proc. Amer. Control Conf.*, 2005, pp. 2989–2994.

[74] A. Bicchi and L. Pallottino, "On optimal cooperative conflict resolution for air traffic management systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 221–231, Dec. 2000.

[75] M. Massink and N. de Francesco, "Modelling free flight with collision avoidance," in *Proc. IEEE Int. Conf. Eng. Complex Comput. Syst.*, Jun. 2001, pp. 270–279.

[76] M. Robb, B. A. White, A. Tsourdos, and D. Rulloda, "Reachability guidance: A novel concept to improve mid-course guidance," in *Proc. Amer. Control Conf.*, 2005, pp. 339–345.

[77] C. Xydeas and C. Brown, "Use of cubic Bézier curves for route planning," in *Proc. 19th Eur. Signal Process. Conf. (EUSIPCO)*, Barcelona, Spain, Aug. 2011, pp. 1786–1789.

[78] E. W. Weisstein. *Spherical Trigonometry*. [Online]. Available: http://mathworld.wolfram.com/SphericalTrigonometry.html

SOHAIL RAZZAQ received the B.Sc. degree in electrical engineering from the University of Azad Jammu and Kashmir in 2004, the M.Sc. degree in computer and network engineering from Sheffield Hallam University in 2006, and the Ph.D. degree in communication systems from Lancaster University in 2012.

He was a Lecturer with the COMSATS Institute of Information Technology from 2007 to 2008, where he has been an Assistant Professor since 2012. His research interests include autonomous systems, network applications, and ICT-based smart grids.

COSTAS XYDEAS received the M.Sc. and Ph.D. degrees from Loughborough University, U.K., in 1974 and 1978, respectively. He was a Research Fellow at Loughborough University in 1977, and then a Lecturer and a Senior Lecturer in 1980 and 1984, respectively, where he directed a M.Sc. course in digital communication systems and led the Research Group in Speech and Image Coding.

He was appointed as a Professor of electrical engineering with The University of Manchester, U.K., in 1987, where he led the Signal Processing Research Group and established the Speech and Image Processing Research Laboratory, the Manchester Multimedia Research Laboratory, and the Manchester Avionics Research Centre. From 1994 to 1998, he was appointed as the Director of Research for the Divisions of Electrical, Mechanical, Aeronautical and Civil Engineering, Faculty of Science, School of Engineering, The University of Manchester.

In 1999, he was appointed as a Professor of signal processing with the Digital Communication Systems Department, Lancaster University, and established initially the Speech/Video/Information Systems Research Group and then the Signal Processing and Intelligent Systems Group. He was appointed as the Head of the Communication Systems Department, and he also served as the Director of undergraduate and postgraduate studies. He retired from his Lancaster University post in 2014.

He is currently a Co-Founder and the Director of Iphestos Systems Ltd., a consultancy SME established in 2009 and specializing in advanced digital signal processing techniques and systems, system's performance modeling, and algorithm development. He has authored of over 130 research papers and holds several U.S. and U.K. patents.

MICHAEL E. EVERETT received the B.A. degree (Hons.) in mathematics from Open University in 1978 and the M.Sc. degree in applicable mathematics from the Cranfield Institute of Technology in 1979.

Following service in HM Forces (R. Signals) he joined BAE Systems (BAES) as an Avionics Systems Analyst in 1979 with interest in MMW radar target recognition algorithms and systems performance analysis. In 1990, he became the Team Leader of the Data Fusion Research Group. In 1995, he was the Head of the Systems Research Department. In 2000, he retired from BAES and took up a part-time Research Fellowship at Lancaster University, where he was involved in target acquisition and collision avoidance algorithm development. In 2009, he co-founded Iphestos Systems Ltd., a systems engineering consultancy specializing in performance modeling and algorithm development.

ANZAR MAHMOOD (M'13–SM'17) received the B.Sc. degree in electrical engineering from the University of Azad Jammu and Kashmir in 2005, the M.Engg. degree in nuclear power from the NED University of Engineering and Technology, Karachi, in 2007, and the Ph.D. degree in electrical engineering from the COMSATS Institute of Information Technology (COMSATS), Islamabad, in 2016.

He was a Full Time Research Fellow for two years. He served as an Assistant Professor with COMSATS for three years. He is currently an Associate Professor with the Department of Electrical (Power) Engineering, Mirpur University of Science and Technology, Mirpur, Pakistan. Before joining the COMSATS, he participated in various projects under the Pakistan Atomic Energy Commission as a Senior Design and an Installation Engineer for six years.

He has published 52 research articles and conference proceedings. His research interests include computational intelligence, smart grids, power systems, and renewable energy.

THAMER ALQUTHAMI (S'05–M'15) received the Ph.D. degree from the Georgia Institute of Technology in 2015. He is currently with the Electrical and Computer Engineering Department, King Abdulaziz University, Jeddah, Saudi Arabia.

He is active in teaching, research, training, and consultation. He has provided consultation to several companies, including Saudi National Grid, the Saudi Energy Efficiency Center, and the King Abdulaziz City of Science and Technology. He is a certified Energy Manager, a certified Energy Auditor, and a certified Local Trainer by the Association of Energy Engineers (AEE). He is a member of Eta Kappa Nu and AEE.