USE OF CUBIC BÉZIER CURVES FOR ROUTE PLANNING

Prof. Costas Xydeas,Colin J. Brown

School of Computing & Communications , Lancaster University,
INFOLAB21 Lancaster University, LA1 4WA, Lancaster, United Kingdom
phone: +(44)(0)1524-510401
email: c.xydeas@lancaster.ac.uk,c.brown7@lancaster.ac.uk

ABSTRACT

We consider the use of cubic Bézier curves for planning UAV routes. The proposed approach allows the user to trade off the length of the solution route with the level of risk/hazard exposure encountered. Exhaustive search is used to place control points on a 2D grid superimposed on the environment. High quality routes are generated using relatively course grids. Comparison is made with the graph theoretic A* technique.

1. INTRODUCTION

There is an increasing deployment of uninhabited air vehicles (UAVs) in many different fields of operation. In general, most operational vehicles demonstrate semi-autonomous behaviour, in so far as, a human operator is required to control or monitor vehicle actions. Considerable research is being targeted at dramatically increasing the level of autonomy of such vehicles therefore reducing the level of "human-in-loop" activity required.

A challenging issue in this field includes route planning. This is the ability of an uninhabited vehicle to plan a path of motion while avoiding both static and dynamic hazards within it's environment.

Our work addresses the routing of Uninhabited Air Vehicles (UAV) operating in a three dimensional environment. However in order to simplify experimentation the problem is reduced to that of a two dimensional (2D) horizontal environment while at the same time all proposed techniques can be easily extended and applied to 3D. Furthermore, current flight management systems separate vertical and horizontal navigation planning so this is seen as a reasonable simplification.

Several approaches to UAV routing can be found in the literature. Examples include Graph Theoretic ([6]), Optimal Control ([3]), Potential Gradient Descent ([5]), Evolutionary Algorithms ([7]) etc. A comprehensive review of approaches is found in [1].

Key in the measure of success of all these routing approaches must be the trade off between the quality of the derived route and the algorithmic complexity of the router.

Graph Theoretic (GT) routing based approaches are quite popular (E.g. A* ([2]) operate on discrete environment spaces and have the potential to produce optimum routes, with respect to a given cost function. However fine grid/state spacing is required to achieve "smooth" routes trajectories and this increases execution time. In contrast, non-smooth routes may not be flyable due to vehicle dynamic constraints.

In GT techniques a state transition is represented as $X_a \rightarrow X_b$. A basic search would calculate a cost from the origin to the new state $G_{new} = f(X_b)$ and, in the case of A*, an estimate of the cost (heuristic) from the new state to goal $H_{new} = h(X_b, X_{goal})$. In addition to assessing the cost of new state X_b, the router must consider the transition path between states. This in effect avoids a solution route cutting across a defined hazard region, a situation that can occur when states X_a, X_b are deemed not to be in a hazard region but the transition path between the states does intercept the hazard. In general the complexity of determining state transition path intercepts is considerably greater than just determining if the new state is within a hazard region.

Furthermore, if hazard regions are considered as totally prohibited, optimal routes are found which tend to "creep" round the hazard perimeter. However, if the route optimality requirement is relaxed and a finite cost is given to a state transition across a hazard rather than a binary yes/no situation, more practical solutions can be obtained. For example, slightly penetrating a radar coverage zone may be preferable to the extra fuel cost of completely avoiding the zone.

In this paper we examine the use of an alternative, polynomial based approach that employs Bézier curves as a method for path planning. This allows for smooth routes to be generated on the basis of user defined risk tolerances while using relatively coarse control point locations, to limit an otherwise prohibitively large search complexity.

Notice that the control points of any Bézier curve form a control polygon or convex hull. Use of this property is explored by ([4]) for de-confliction of multiple vehicles.

2. SYSTEM DESCRIPTION - BÉZIER CURVES

In this paper we consider the use of Bézier curves for route planning in general and employ third degree Bézier

curves in particular. A degree n Bézier curve is defined with n+1 control points. The formal Bézier curve definition is:

$$\mathbf{P}(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i \mathbf{P}_i = \sum_{i=0}^{n} \mathbf{B}(t)\mathbf{P}_i \quad (1)$$

where $B(t)$ is known as the Bernstein polynomial (*blending function*) and P_i is a control point $i = 0, 1, \ldots, n$. The resultant curve may be seen as a weighted *blend* of the control points. The curve is guaranteed to pass through the first and last control points.

Thus a third degree Bézier curve is defined with 4 control points P_0, P_1, P_2, P_3. We consider the first control point P_0 to be the start and P_3 the end or goal of a route. The location of these points is therefore fixed. The remaining control points i.e. positions P_1, P_2 are manipulated to generate a candidate Bézier curve. A computationally efficient scheme is used to evaluate $\mathbf{P}(t)$ rather than direct use of equation 1.

Using MATLAB as the simulation platform, a 2D unit grid is used to define a the operational space $(0 \leq x, y \leq 1)$. Obstacles are represented by H non-overlapping circles having random centres and constrained radii lengths.

A Bézier curve may be scaled and/or rotated by application of a transformation matrix to the control points. This useful property of Bézier curves would enable the unit grid results to be scaled to any desired dimensions (kilometres,nautical miles etc.). The ratio of total obstacle area and unit grid area is a measure of hazard density. We believe our experiments use a representative hazard density to that of real world airspace.

2.1 Cost Function

The cost function C of a given Bézier curve based route is defined here in terms of two types of "risk" components PL and RE, although this can be easily extended to include additional risks. Thus:

$$\mathbf{C} = \frac{1}{2PL_{max}} (PL + RE) \quad (2)$$

PL is the total route length risk, RE is the hazard exposure risk which is the total length of the route which is within hazard regions. RE is formed as a sum of each hazard incursion length RE_i over all H hazards in the scenario.

$$RE = \sum_{i=1}^{H} RE_i$$

PL_{max} is an upper bound on route length. This is used to normalise both PL and RE to unit values. The factor of 2 in Equation 2 yields an overall cost in the range $0 < \mathbf{C} \leq 1$.

The route path length is evaluated numerically as the line integral of $\mathbf{P}(t)$ over the parameter interval $(0 \leq t \leq 1)$.

The risk exposure RE_i is calculated by finding the roots of an expression developed by inserting $\mathbf{P}(t)$ into the equation defining the i_{th} circular hazard. The solutions (if any) to this equation are in terms of values of parameter t. Taking into account the possibility of either or both the start and goal control points being themselves within a hazard the path length of any intersection can be evaluated by line integration using the solution roots as limits on parameter t. $PL_{max} = 2$ is used in these experiments. This is consistent with the longest Bézier curve possible on the unit grid.

2.2 Exhaustive Search

The initially continuous unit space environment, within which the optimal location of control points is to be estimated, is discretized into a $N \times N$ grid. Thus each point in the grid represents a possible control point location. Holding the start and goal control points P_0, P_3 fixed, the number of locations for P_1 and P_2 is:

$$\frac{2N^2!}{2!(N^2-2)!} = \frac{N^2!}{(N^2-2)!}$$

The factor of two in the left hand numerator accounts for the fact that two Bézier routes defined as P_0, P_1, P_2, P_3 and P_0, P_2, P_1, P_3 will, in general, not be the same. Thus the proposed system employs an exhaustive search approach to define the minimum cost function route, with the possibility of also using a maximum exposure to risk threshold.

3. EXPERIMENTATION AND RESULTS

3.1 Grid Resolution

Figure 1 shows route generation results for an example scenario consisting of four hazard regions in a discrete $N \times N$ grid space, where $3 \leq N \leq 5$. Figure 2 illustrates route results for the same scenario with $6 \leq N \leq 8$. The figures additionally indicate an A* generated route using a radial (= 0.1 units) node expansion scheme with angle deviations of $-10^\circ, -5^\circ, 0^\circ, +5^\circ, +10^\circ$ about the current heading. The A* implementation uses a total cost function $f(\mathbf{x})$ at node/state $\mathbf{x}$ given by $f(\mathbf{x}) = g(\mathbf{x}) + \gamma.h(\mathbf{x})$, where $g(\mathbf{x})$ is the distance from the origin travelled to node/state $\mathbf{x}$ and $h(\mathbf{x})$ is the heuristic cost i.e. the Euclidean distance from $\mathbf{x}$ to the goal. γ has been defined experimentally to the value $\gamma = 1.25$. This value provides a bias towards depth-first type of search and yielded significant improvements in execution time for our experimental scenarios. Note that the A* routes are not as smooth as the Bézier derived solutions.

From the figures and many others produced using different environment configurations, it can be seen that all Bézier routes are represented by smooth trajectories. It has been also observed that, in several cases, solution routes are almost coincident with others obtained from different value of N. Table 1 summarises run times and

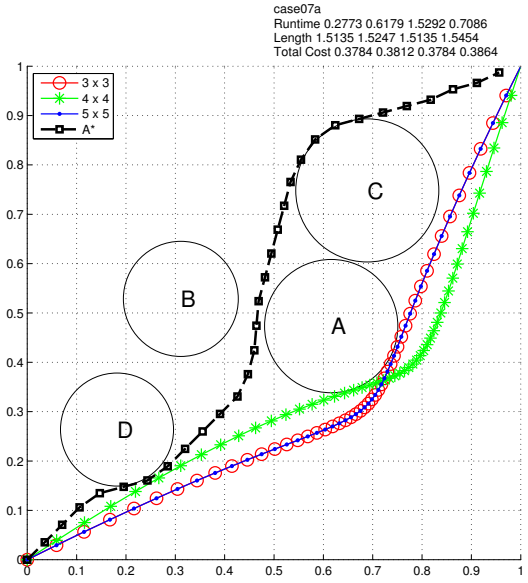

Figure 1: Grid $3 \leq N \leq 5$

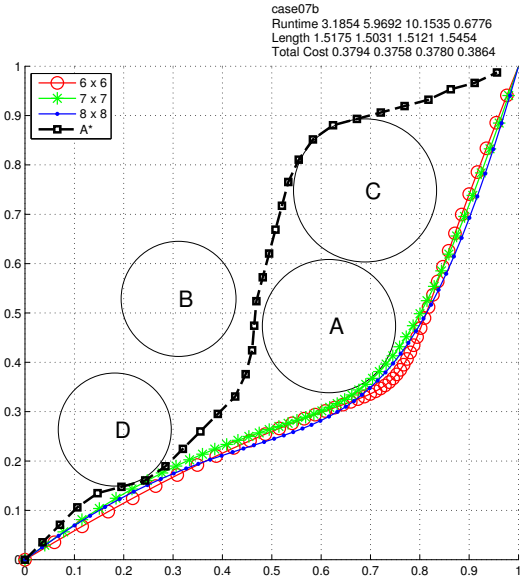

Figure 2: Grid $6 \leq N \leq 8$

route cost C values for the different values of N used in the scenario shown in Figures 1 and 2. Note that experiments were performed over 100 randomly selected environment scenarios, for each 3, 4 and 5 hazards, and results represent averages accordingly. Run times are in seconds with runtime relative to A* shown in parenthesis. Furthermore, run times are averaged over 10 executions of each test case in order to account for fluctuations in processor availability. The runtime value for A* is also shown in Table 1 for comparison. Furthermore, it is seen that the best, i.e. minimum cost route does not necessarily result from the finest grid spacing. In general good route solutions are obtained for relatively small values of N, i.e; $N < 5$, which means that the proposed route generation approach is computationally efficient and fast.

N	Runtime	Cost C
3	0.277 (0.4)	0.378
4	0.618 (0.9)	0.381
5	1.529 (2.2)	0.378
6	3.185 (4.7)	0.379
7	5.969 (8.8)	0.376
8	10.154 (15.0)	0.378
A*	0.709 (1.0)	0.386

Table 1:

3.2 Risk Trade-Offs

Results previously presented show the minimum cost **C** route for a given grid resolution. Alternatively the exhaustive search approach allows the two components of risk to be traded off. For example, we may accept a small amount of hazard intrusion in return for an overall shorter route. Numerous such trade-offs are possible with this technique. The Bézier router therefore can return a family of routes that satisfy user input constraints.

To demonstrate this approach, while using a grid resolution of 4×4, we identify the route with the largest hazard intrusion value RE i.e. RE_{max}. A subset of routes with exposure values within a defined range in relation to this value can then found. The lowest cost **C** of this subset is then selected. Figure 3 shows examples for the range 5-10%,20-30% and 40-50% of RE_{max}. The overall minimum cost route is also shown.

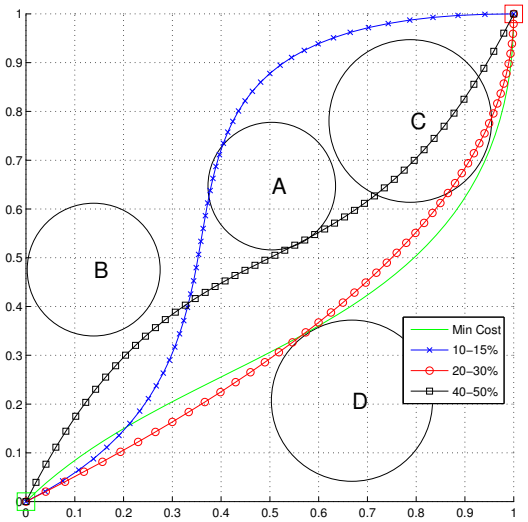

Figure 3: Routes within a range of risks

3.3 Scenario Complexity

Figure 4 presents a scatter plot of the logarithm of execution times for all test scenario cases, as applied to the Bézier method (with grid sizes $N = 3,4,5$) and the A* approach. Test cases #1-#100 represent three hazard scenarios ($H = 3$), #101-#200 four hazards ($H = 4$) and #201-#300 five hazards ($H = 5$). Annotation within the plot makes this clearer.

Figure 4 shows that while A* is, in general, faster than the Bézier method, the latter is more scenario independent. For the Bézier router with $N = 3,4$ the distribution of execution times is almost flat for all the test cases. There is greater deviation in the $N = 5$ test cases. A detailed examination of Figure 4 also shows a number of A* cases with run times very much greater than the mean and, in certain cases, greater than any value obtained from the Bézier router. This run time dependency of A* on the configuration of the environment is characteristic for all N values i.e. $3,4,5$. The worst case execution time for any of the test cases and methods is claimed by A* at 12.75 seconds with the Bézier method ($N = 3$) taking only 0.21 seconds for the same test case. Table 2 presents average run times $\bar{t}$, standard deviations $\sigma(t)$ and the ratio of these statistics.

Method	$\bar{t}$	$\sigma(t)$	$\bar{t}/\sigma(t)$
3×3	0.171	0.022	0.129
4×4	0.557	0.073	0.132
5×5	1.381	0.185	0.134
A*	0.400	1.096	2.738

Table 2:

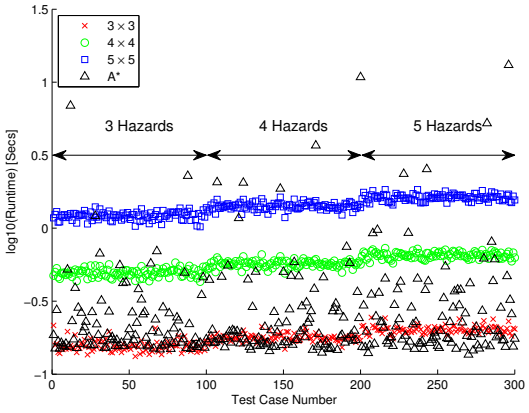

Figure 4: Scatter Plot of log10(Runtime) for all Test Cases

4. CONCLUSION

The use of cubic Bézier curves in the generation of routes for a UAV has been considered in this paper and a relatively low complexity router is proposed. Computer simulation based experimentation has clearly shown that, while the search space for the location of control point is kept coarse and therefore search complexity is low, the generated routes are of "high quality" in terms of both smoothness and cost. Furthermore when compared to A*, Bézier based run times are confined within narrow distributions about the mean and they are also independent of operating environment scenario characteristics/complexity.

The proposed router has the distinct advantage of defining routes in terms of two control point locations and therefore can also be viewed as offering "compression" of the route specification. In addition, the user can also obtain solutions representing trade-offs between different types of risks and the level of risk that is to be tolerated. Further analytic constraints could be applied, such as limiting trajectory curvature values, thus making this technique particularly attractive in relation to the large variability of possible types of UAV platforms.

Note that the proposed approach is easily amenable for extension to three dimensions and the inclusion of 3D hazard primitives such as ellipsoids, polyhedra and cylinders.

REFERENCES

[1] D. Gu, I. Postlethwaite, and Y. Kim. A comprehensive study on flight path selection algorithms. In *Target Tracking: Algorithms and Applications, 2006. The IEE Seminar on (Ref. No. 2006/11359)*, pages 77–90, 2006.

[2] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, 4(2):100–107, 1968.

[3] T. Kinoshita and F. Imado. A study on the optimal flight control for an autonomous UAV. In *Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on*, pages 996 –1001, 2006.

[4] M. I. Lizarraga and G. H. Elkaim. Spatially deconflicted path generation for multiple UAVs in a bounded airspace. In *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pages 1213 – 1218, 2008.

[5] N. J. Nilsson. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann; I.S.ed edition, 1998.

[6] K. Tulum, U. Durak, and S. K. Yder. Situation aware UAV mission route planning. In *Aerospace conference, 2009 IEEE*, pages 1 –12, 2009.

[7] C. Zheng, L. Li, F. Xu, F. Sun, and M. Ding. Evolutionary route planner for unmanned air vehicles. *Robotics, IEEE Transactions on*, 21(4):609 – 620, Aug. 2005.