Proceedings of the American Control Conference
Chicago, Illinois • June 2000

Path Planning for UAVs

Scott A. Bortoff[1]
United Technologies Research Center
411 Silver Lane, E. Hartford CT 06108
bortofsa@utrc.utc.com

Abstract

In this paper, a two step path-planning algorithm for UAVs is proposed. The algorithm generates a stealthy path through a set of enemy radar sites of known location, and provides an intuitive way to trade-off stealth versus path length. In the first step, a suboptimal rough-cut path is generated through the radar sites by constructing and searching a graph based on Voronoi polygons. In the second step, a set of nonlinear ordinary differential equations are simulated, using the graph solution as an initial condition. The ODEs describe the dynamics of a set of virtual masses located in a virtual force field. The virtual forces push the masses away from the radars and toward one another. The ODEs are simulated to find a locally exponentially stable equilibrium solution, which is interpreted as the optimal path. A simulation is provided to illustrate the idea.

1 Introduction

Among the many open issues to be addressed in the development of UAVs is that of *path planning.* A path planning algorithm computes a trajectory from the UAV's present location to a desired future location, e.g. a target.

A good path planning algorithm for UAVs must possess several important attributes, making its design a multiple-objective optimization problem. First it must compute a *stealthy* path, steering the aircraft and its radar signature around known enemy radar locations. This is difficult because no aircraft scatters or reflects radar radiation uniformly in all directions. Rather, radiation is radiated more strongly in some directions, and less strongly in others. The path planning algorithm should take advantage of any "spikes" in the signature and point them away from known enemy radar locations. Second, generated trajectories should be of minimal length, subject to the stealthy constraint, and also satisfy the aircraft's dynamic constraints. Third, the path-planning algorithm must be compatible with the cooperative nature envisioned for the UAV. A typical mission might involve multiple UAVs attacking single, well-defended target. The path-planner would be a component — a subroutine — of the hybrid control system that ensures all UAVs arrive simultaneously. And finally, path-planning algorithms are expected to be coded in software that runs on an airborne processor. Thus, they must be computationally efficient and "real time," enabling the UAV to re-plan its trajectory should an unforeseen threat arise. This last characteristic should not be dismissed as an implementation issue to be addressed solely with increased computer power.

In this paper, we propose a two-step method to generate paths through the hostile territory illustrated in Figure 1. In the first step, a simple graph is constructed and searched, generating a rough-cut suboptimal path. This solution is used to initialize the second step, which uses virtual force fields to improve upon the graph solution. The approach produces a stealthy path with minimal path length, and does not suffer from the curse of dimensionality, making it suitable for real-time implementation.

2 Step 1: Voronoi Graph Search

A graph is a set that contains vertices and edges. Graphs can be used to solve the UAV path planning problem, just as they do for many similar robotic path-planning problems, by assigning vertices to discrete points of state space, connecting them appropriately with edges, assigning weights (costs) to each edge, and then searching the graph for an optimal trajectory using one of several well-known algorithms. The dynamic constraints of the aircraft are roughly discretized in this approach by the edge assignment. For example, each vertex might represents a discrete position (x, y, z) and orientation (ϕ, θ, ψ) in space. The vertices are connected by an edge only if the aircraft can actually fly between those points.

Unfortunately, this approach is suffers from the curse of dimensionality, making real-time implementation a challenge. One way address this computational complexity is to use a sequence of graphs. We start with a relatively course graph, one with a small number of

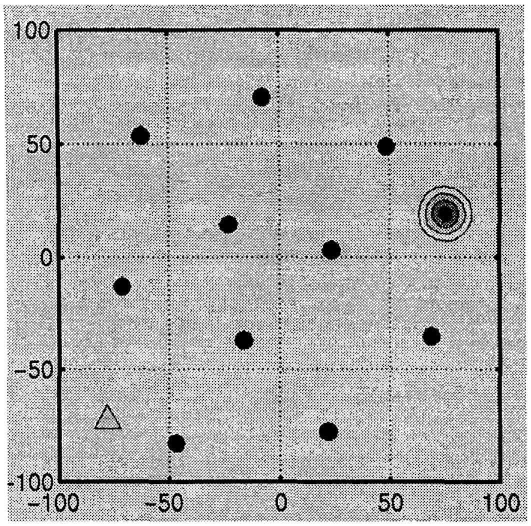

Figure 1: The path planning problem: Find the path from the UAVs present position (triangle), to the target (concentric circles) that minimizes a weighted cost that penalizes total path length and also distance from the radar sites (small circles).

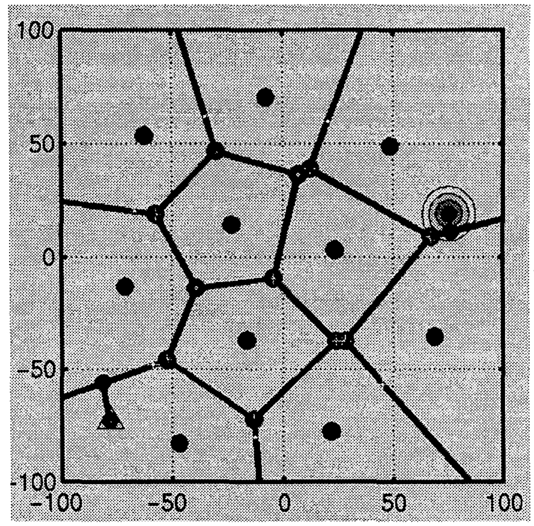

Figure 2: Constructing the Voronoi graph begins with knowledge of the radar sites (small circles), the UAV (triangle) and target (concentric circles). Triplets of radar sites are selected such that the unique circle that passes through them contains no other radar sites. These triplets form the Delaunay triangulation, and the center of each circle is a Voronoi point. Two Voronoi points are then connected to form an edge if their associated Delaunay triangles share an edge. The Voronoi graph is completed by extending rays from the edge Voronoi points midway through the appropriate Delaunay triangle edge. The target and UAV are then attached to the graph.

vertices, and search it for an optimal solution. We then build a new graph in a neighborhood of this optimal solution, and search it for a new optimal solution. The second graph would have a higher density of vertices and edges. This procedure can be repeated. In this way, a detailed level of quantization is achieved around the final trajectory

Taking this approach to its extreme, we might begin by constructing the simplest possible graph that captures the very essence of the problem. Such a graph can be constructed using a Delaunay triangulation and its geometric dual, Voronoi polygons. This procedure, which is begins with complete knowledge of the number and location of each radar site, as illustrated in Figure 1. For every triplet of radar sites, there exists a unique circle that passes through all three. Consider only those triplets whose circle does not enclose any other radar sites, as shown in Figure 1 (bottom). The set of all such triplets is called the Delaunay triangulation, and the centers of the circles are called Voronoi points. We may now construct a graph by defining the vertices as the Voronoi points. Edges are drawn to connect two Voronoi points if and only if their associated Delaunay triangles share an edge. By drawing all such edges, we construct the Voronoi diagram or graph. The edges of the Voronoi diagram have the property that they are equidistant from pairs of radar sites.

If we consider the Voronoi diagram to be a graph, with vertices being the Voronoi points and edges being the connecting segments, then we can assign weights to each edge just as we would for any graph. Weights are defined as a function of edge length and probability of detection by neighboring radars. Once the weights are assigned, the complete graph can be searched for the optimal path by dynamic programming. This produces an optimal path from the set of Voronoi segments which is the simplest path through the radar sites, in the sense that it "tells" the UAV which pairs of radar sites to fly between. While this approach might provide an overly simplified, coarse flight path, it is useful as an initial condition for other methods that can capture more of the detail inherent in the path planning problem, such as a more refined graph or the optimal control approach outlined in the next section.

3 Step 2: Virtual Forces

For the second step of the overall approach, a representation of the path is generated via the steady-state equilibrium solution to a Lagrangian mechanical system driven by virtual forces. In this method, a UAV path is represented by a "chain of point masses." The masses are connected to one another by springs and

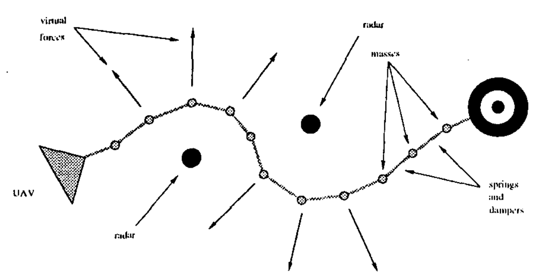

Figure 3: The chain-of-masses being acted on by both the spring restoring forces and the virtual forces pushing from each radar site.

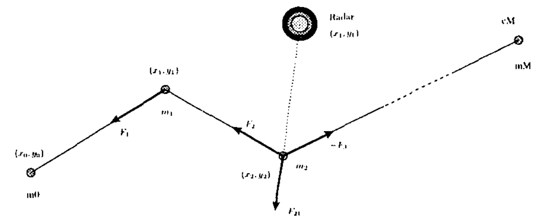

Figure 4: The chain-of-masses idea, showing the two spring forces ($\boldsymbol{F}_1$ and $-\boldsymbol{F}_2$ and one virtual "radar" force ($\bar{\boldsymbol{F}}_{21}$) acting on mass $j = 2$.

dampers, as shown in Figure 3. The springs exert a contractive force, and act to reduce the length of the chain. One end of the chain is attached to the UAV location, while the other is attached to the target location.

Ignoring the radar sites for now, if this system of springs, dampers and masses is initialized at *any* initial configuration, it will evolve in time according to the governing physics, and eventually converge to its potential energy minimum. Of course, because the only forces between the masses are due to the springs and dampers, this minimum energy configuration is a straight line of masses: The dampers remove all of the kinetic energy. If there were no enemy radar sites nearby, then this would be an optimal flight path between the UAV location and its target.

Now suppose there are enemy radar sites nearby the UAV. The key idea here is to force the chain of masses away from radar sites by using a virtual force field. We first consider the case of a uniform radar signature. In this case, the signal strength of the return echo is inversely proportional to distance between the UAV and the radar site to the fourth power. So to force the path away from the radar site, assume that each radar site establishes a repulsive force field which acts to push away each mass according to an inverse-to-the-fourth law (1/distance^4). The force acts along the straight line containing the mass and radar site. Thus, the total force acting on each mass is the vector sum of all the forces acting from the radar sites, plus the two spring forces acting from neighboring masses. When the dynamics of this system is initialized in *any* configuration, it will eventually converge to a potential energy minimum, because the dampers will remove all of the kinetic energy.

This minimum energy configuration is a weighted sum of path length and the average distance from the radar sites. This is because the spring forces acting on each mass tend to minimize length, while the radar forces tend to maximize distance from the radars. By weighting these forces appropriately, an optimal (in the sense of minimizing potential energy) path can be generated. Once the system has converged, the path is defined as the sequence of way points defined by the steady-state mass locations, connected by straight line segments. The idea is illustrated in Figures 3 and 4.

3.1 Uniform Radar Signatures

Referring to Figure 4, the equations of motion for the uniform-radar signature case are derived as follows. Assume that we have $M + 2$ masses, indexed by j $(0 \leq j \leq M+1)$, where the first mass ($j = 0$) is located at the UAV location and the last mass ($j = N + 1$) is located at the target location. Let m_j denote the mass of mass j. Let (x_j, y_j) denote the (x, y)-coordinate of mass j, so (x_0, y_0) is the UAV Cartesian coordinate at time t_0, and (x_M, y_M) is the target Cartesian coordinate, i.e., the desired UAV location at time t_f. Assume the springs between the masses are linear with a spring constant of κ, and let the dampers between each of the masses have a damping constant of b.

With this notation, the distance between each of the $M + 2$ masses is

$$d_j = \sqrt{(x_j - x_{j-1})^2 + (y_j - y_{j-1})^2} \qquad (1)$$

for $(1 \leq j \leq M+1)$. Denote the normal vector pointing from mass j to mass $j - 1$ as $\boldsymbol{n}_j$, so that

$$\boldsymbol{n}_j = \begin{bmatrix} (x_{j-1} - x_j)/d_j \\ (y_{j-1} - y_j)/d_j \end{bmatrix} \quad \text{for} \quad (1 \leq j \leq M+1).$$

Then the two spring restoring forces that act on mass j $(1 \leq j \leq M)$ are given by Hooke's law:

$$\begin{aligned} \boldsymbol{F}_j &= \kappa d_j \boldsymbol{n}_j \\ \boldsymbol{F}_{j+1} &= -\kappa d_{j+1} \boldsymbol{n}_{j+1}. \end{aligned}$$

The two linear viscous damping forces that act on mass j, which we denote with a $\widehat{\ }$, are similarly expressed as

$$\begin{aligned} \widehat{\boldsymbol{F}}_j &= b \cdot \left(\begin{bmatrix} \dot{x}_j \\ \dot{y}_j \end{bmatrix} \cdot \boldsymbol{n}_j - \begin{bmatrix} \dot{x}_{j-1} \\ \dot{y}_{j-1} \end{bmatrix} \cdot \boldsymbol{n}_j \right) \cdot \boldsymbol{n}_j \\ \widehat{\boldsymbol{F}}_{j+1} &= b \cdot \left(\begin{bmatrix} \dot{x}_{j+1} \\ \dot{y}_{j+1} \end{bmatrix} \cdot \boldsymbol{n}_{j+1} - \begin{bmatrix} \dot{x}_j \\ \dot{y}_j \end{bmatrix} \cdot \boldsymbol{n}_{j+1} \right) \cdot \boldsymbol{n}_{j+1}, \end{aligned}$$

where the $\cdot$ denotes the dot product.

Finally, the "virtual force" acting on mass j ($1 \leq j \leq M$) from radar site k ($1 \leq k \leq N$) is defined to obey an "inverse-squared-squared law." If each of the N radar sites are located at $(\bar{x}_k, \bar{y}_k)$, then the distance from mass j to radar site k is

$$\bar{d}_{jk} = \sqrt{(x_j - \bar{x}_k)^2 + (y_j - \bar{y}_k)^2 + h^2}$$

where h is the height of the UAV, assumed to be constant. If we denote the normal vector pointing from radar site k to mass j as

$$\bar{n}_{jk} = \left[\begin{array}{c} (x_j - \bar{x}_k)/\bar{d}_{jk} \\ (y_j - \bar{y}_k)/\bar{d}_{jk} \end{array} \right],$$

then we define the virtual force acting on mass j from radar k as

$$\bar{\boldsymbol{F}}_{jk} = \frac{Q}{\bar{d}^4_{jk}} \bar{n}_{jk}$$

where Q is a constant design parameter that represents the trade-off between stealth and path length. With this, the equations of motion for each mass expressed in Cartesian coordinates is just given by Newton's law:

$$m_j \left[\begin{array}{c} \ddot{x}_j \\ \ddot{y}_j \end{array} \right] = \boldsymbol{F}_j + \boldsymbol{F}_{j+1} + \widehat{\boldsymbol{F}}_j + \widehat{\boldsymbol{F}}_{j+1} + \sum_{k=1}^{N} \bar{\boldsymbol{F}}_{jk},$$

which, expressed in Cartesian coordinates is

$$\begin{aligned} m_j \ddot{x}_j &= \underbrace{\kappa(x_{j-1} - x_j) - \kappa(x_{j+1} - x_j)}_{\text{spring forces}} \\ &+ \underbrace{b(\dot{x}_{j-1} - x_j) - b(\dot{x}_{j+1} - x_j)}_{\text{damping forces}} \\ &+ \underbrace{\sum_{k=1}^{N} \frac{Q}{((x_j - \bar{x}_k)^2 + (y_j - \bar{y}_k)^2 + h^2)^{5/2}}}_{\text{virtual forces}} \quad (2) \end{aligned}$$

for $1 \leq j \leq M$. (A similar equation holds for the y-direction.) Note that the equations of motion for masses $j = 0$ and $j = M + 1$ are not expressed here because they are assumed to be fixed in space.

Now, to find a path, we first triangulate the region using the graph theory approaches outlined in Section 2. This produces a sequence of straight-line paths, which we use to initialize the virtual force approach. This is done by placing the M masses uniformly along these straight-lines, and then simulating the equations of motion (2) until the solution reaches equilibrium. The parameter Q is set to trade-off stealth versus path-length. If $Q = 0$, then no penalty is placed on stealth, the nonlinear terms in (2) vanish, and the masses will converge to the global straight-line equilibrium that minimizes the potential energy in the springs. For large Q, however, the masses will be pushed away from the radar sites, bending around them, reaching a local equilibrium that represents a trade-off between minimal path length and average distance from the radars. The result of a typical Matlab simulation is illustrated in Figure 5.

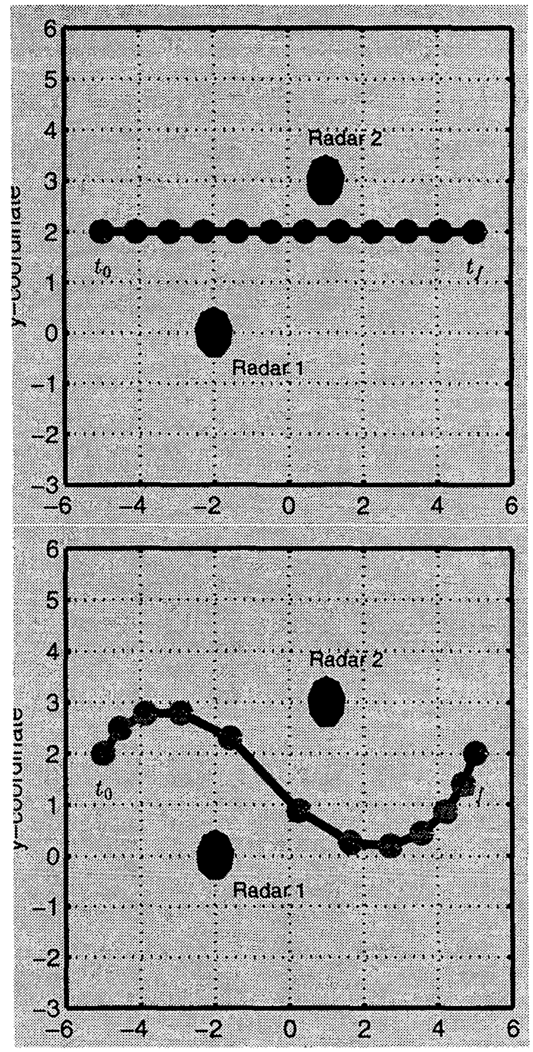

Figure 5: At top, the 10 masses ($M = 11$) are initialized along a straight line, which would be a segment from the Voronoi graph. The system of equations (2) is simulated and allowed to reach equilibrium, shown at bottom. For this simulation, $b = 1$, $\kappa = 1$, $Q = 50$, $m_j = 1$, and the simulation time is 20s.

Several remarks are in order. First, note that the right-hand sides of (2), although nonlinear, are globally Lipschitz. This means that the solution to the differential equations will exist for all time. Moreover, note that the virtual force never larger than $1/h^4$, i.e., it "saturates" for large distances $\bar{d}_{jk}$. This means that the linear spring terms will dominate at large distances

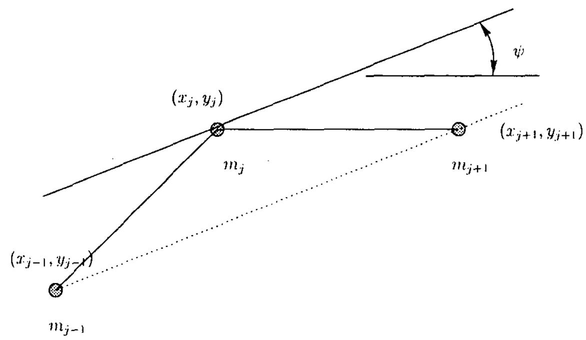

Figure 6: Assigning an orientation ψ_j to mass m_j using the approximate derivative, to incorporate the effects of a non-uniform radar signature.

from the radars, as intuition would suggest: The path should not be affected if a radar site is very far away. From a numerical simulation point-of-view, the non-linearity is "soft" (not severe), which makes for easy simulation of (2). This is important from the real-time implementation point-of-view.

3.2 Non-Uniform Radar Signatures

Equations (2) must be modified to incorporate the effect of a non-uniform radar signature. For this, each mass m_j must be assigned an orientation in space, because the radar signature is a function of at least yaw ψ (and possibly roll ϕ) of the aircraft, measured relative to an inertial frame. The yaw (azimuth) orientation angle ψ_j can be defined in a discrete way for mass j by computing the angle between the x-axis of an inertial frame and the straight line that passes through mass $j-1$ and $j+1$:

$$\psi_j = \arctan(y_{j+1} - y_{j-1},\, x_{j+1} - x_{j-1}),$$

This is essentially the discrete approximation of the derivative of a curve, as shown in Figure 6. Then the aircraft's signature function $s(x_j, y_j, \psi_j, \bar{x}_k, \bar{u}_k)$, which depends on the relative orientation of the aircraft relative to each radar location can be evaluated, and the forces in (2) can be replaced by $Q \cdot s(x_j, y_j, \psi_j, u_j, \bar{x}_k, \bar{u}_k)$.

This modification makes the virtual force stronger when a radar lobe is pointing toward a radar station. But, the virtual force is still bounded by $Q \max s/h^2$ because s is a globally bounded function of its arguments. Thus, solutions to (2) still exist for all time, and the modified mechanical system will again converge to a potential energy minimum. Since the virtual force is large when a lobe is pointed to a radar station, the system should converge such that the lobes point away from the stations, because the potential energy is less here. However, because the system is nonlinear, convergence will be to a local potential energy minimum; a unique global solution can not be expected to exist.

Several simulations of the non-uniform case have been conducted, and the results can be obtained from the author [1]. The final "picture" for the non-convex case is similar to Figure 5.

Note that the Lagrangian mechanical system produces a path that is the optimal solution in a calculus-of-variations sense. The potential energy stored in the springs, $\sum_{k=1}^{N} \kappa d_i$ is just κ times the path length. When $Q = 0$, the radar weight is zero, the method produces a straight line, just as a calculus of variations approach would. When $Q \geq 0$, some potential energy is "stored" in the potential field generated by the radar stations. Thus, what is minimized is a weighted sum of path length (energy stored in the springs) and distance from the radars. This is a discrete-space (lumped) version of what is being solved by the optimal control problem.

Of course, the path planning problem could be posed as an optimal control problem from the start [1]. However, computing a solution to the optimal control problem involves the numerical solution to a two-point boundary value problem, which is much more difficult to solve than the initial value problem solved here.

4 Conclusions

In this paper, we have presented an efficient method to generate optimal UAV flight paths over hostile territory. The approach can trade-off stealth versus path length, and is suitable for real-time implementation. It is also flexible, in that dynamic environments and pop-up threats are easily incorporated.

References

[1] S. A. Bortoff, "Path planning for unmanned air vehicles," tech. rep., AFRL/VAAD, Wright-Patterson AFB, 1999.

[2] V. Jurdjevic, *Geometric Control Theory.* New York: Cambridge, 1997.

[3] H. Sagan, *Introduction to the Calculus of Variations.* New York: McGraw-Hill, 1969.

[4] S. M. Roberts and J. S. Shipman, *Two-Point Boundary Value Problems: Shooting Methods.* New York: Elsevier, 1972.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms.* Cambridge, MA: The MIT Press, 1990.

[6] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control.* Waltham, MA: Ginn and Company, 1969.