Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Engineering 29 (2012) 4198 – 4203
Procedia Engineering
www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

Path Planning of UAV Based on Voronoi Diagram and DPSO

Han Tong*, Wu Wen chao, Huang Chang qiang, Xuan Yong bo

Engineering College, Air Force Engineering University, Xi'an 710038, Shanxi, China

Abstract

This paper presents a solution strategy for achieving cooperative timing among teams of vehicles. The initial paths are constructed based on Voronoi diagram and based on that, the smoothing paths are designed for the Waypoint Path Planner (WPP). Then it considers the cooperative timing problem and proposes a particle swam optimization algorithm for simultaneous attack. Simulation results show that the approach is of high efficiency in multi-UAVs cooperative timing problem.

Keywords: unmanned aerial vehicle(UAV); path planning; Voronoi Diagram; particle swam optimization

1. Introduction

With the recent development of autonomous technologies, unmanned aerial vehicle(UAV) plays an important role to a wide variety of missions, especially in military applications, such as border patrol, target prosecution and et.al. However they can carry only limited number of resources that deplete with use. Compared to a single UAV with limited payload, it is necessary to deploy a team of UAVs whose aggregate munition capability is sufficient to destroy all the targets in the region of interest0, so a team of UAVs are required to carry out the mission while a sub-team of UAVs (a coalition) may be required to attack a target simultaneously. There are numerous technical challenges to overcome to develop a viable cooperative planning method for a distributed team of UAVs. Firstly, the strategy should facilitates cooperative timing and meet the requirement of timing sequence constraints; Secondly, the complexity of the cooperative path planning problem is increased by the possibility of a changing environment, so the

*Han Tong. Tel.: +86-029-84787555; fax: +86-029-84787555.
E-mail address: guigui21cn@163.com.

doi:10.1016/j.proeng.2012.01.643

algorithm should be executed properly . Finally, trajectories should be provided by the path planner and it must be within the dynamic capabilities of the UAV.

One approach for the first requirement is to apply timing constraints to the task-assignment problem. Mixed-integer linear programming (MILP) is used to solve tightly coupled task-assignment problems with timing constraints in Refs.0. The advantage of this method is that it yields the optimal solution for a given problem and the disadvantages are the complexity of problem formulation and the computation.

For the second problem, Refs.0presents experimental results for two cooperative timing missions carried out using a team of three miniature air vehicles (MAVs). Using a cooperative timing algorithm based on coordination functions and coordination variables, the MAV team executed a series of simultaneous arrival and cooperative fly-by missions.

For the last problem, Refs.0Using particle swarm optimization, alternate paths are generated using B-spline curves, optimized based on the three defined objectives. The resulting paths can be optimized with a preference toward maximum safety, minimum fuel consumption, or target reconnaissance Refs.0present a constant factor approximation algorithm for the allocation problem with the motion of the vehicles satisfies a nonholonomic constraint.

2. Route Generation and smoothing Based on Voronoi Diagram

To avoid the dangerous threat regions and save fuel and time, it is required to construct an optimal path from the start location to the target location. The Voronoi-diagram is adopted abroad as it can be constructed expediently and that especially suited to real time computing[6,7] and one can get some suboptimal initial paths using this method,.

Let $P=\{p_1, p_2, \cdots, p_i, \cdots, p_n\}$ be a set of points in a two-dimensional Euclidean plane. These points are called sites. A Voronoi diagram decomposes the space into regions around each site, such that all points in the region around p_i are closer than to any other point in P . Based on this description, the Voronoi region $V(p_i)$ for each p_i is expressed as equation 1:

$$V(p_i)=\{x:|p_i - x| \le |p_j - x|, \forall j \ne i\} \tag{1}$$

Region $V(p_i)$ consists of all points that are closer to p_i than to any other site. The region set of all sites form the Voronoi Diagram $V(P)$. For a battle area having M threats, the Voronoi graph partitions the battle area into M convex regions. Each region contains one threat and any point within a region is closer to the enclosed threat than to any other. The edges of the Voronoi graph represent lines that are equidistant from the two closest neighboring threats. Therefore, the graph edges maximize the distance from the two closest threats. The Voronoi graph also contains initial and final locations within cells to ensure that threats will be avoided when joining and leaving the graph. The construction of Voronoi graph has been implemented in MATLAB which offers the function voronoi.m.

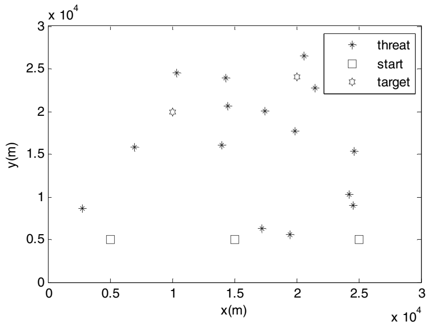

Fig.1 Initial conditions for Voronoi diagram

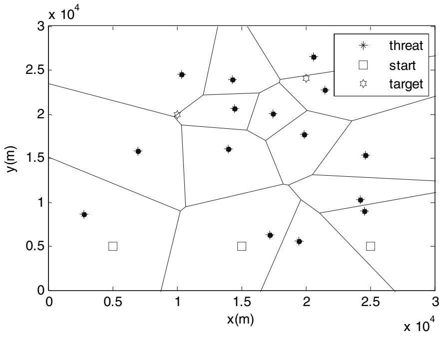

Fig.2 Construction for Voronoi diagram

Initial known threat locations and Voronoi-diagram are described in Fig. 1. The Voronoi-diagram consists of nodes and cells. The edges of the cells are available paths to the nearest node to the target positions. Now we have to search the shortest and safest path to go to the nearest node to the target positions. Before this work, the costs of each Voronoi edge should be decided. The total cost of each edge consists of both length and exposure cost. The cost of the edge is given by:

$$J(w_i) = \lambda_1 \sum_{k=1}^{n} f(a_i, p_k) + \lambda_2 L(a_i) \tag{2}$$

$$f(w_i, p_k) = p_{k\_v} / (L(w_i, p_k)^4) \tag{3}$$

$$L(w_i, p_k) = \sqrt{[(w_{i1}^x + w_{i2}^x)/2 - p_k^x]^2 + [(w_{i1}^y + w_{i2}^y)/2 - p_k^y]^2} \tag{4}$$

$$L(w_i) = \sqrt{(w_{i1}^x - w_{i2}^x)^2 + (w_{i1}^y - w_{i2}^y)^2} \tag{5}$$

Where w_i denote the route segment in Voronoi diagram, w_{i1} , w_{i2} represent the two vertex for w_i respectively. p_k is the threat point and p_{k_v} is the threat power value. λ_1 and λ_2 are weights value and $L(w_i, p_k)$ is defined as the length from w_i to p_k.

As shown in Fig.2, the Waypoint Path Planner (WPP) produces waypoint paths for each vehicle that are low in cost and satisfy the requirement of safety. The objective of the dynamic trajectory smoother (DTS) is to smooth the straight-line waypoint paths into time-parameterized trajectories that are flyable by the UAV. The waypoint paths have been chosen to satisfy timing constraints; therefore the trajectories must be smoothed so that the resulting path length is identical to the waypoint path length. In addition, because individual and team objectives are based on the cost of the waypoint paths, the smoothed trajectory must deviate as little as possible from the waypoint paths produced by the WPP.

In this paper we assume that the UAV is flying at constant altitude and is equipped with trajectory-tracking capability. The input to the DTS is a constant feasible velocity $v_i^c \in [v_{\min}, v_{\max}]$ and a constant waypoint path: $W_i = \{w_{i1}, w_{i2}, \cdots, w_{ip}\}$,Where $w_{ij} \in \mathbb{R}^2$ denote the waypoints expressed in inertial coordinates. The DTS is given by the differential Eq.(6):

$$\dot{x}_{ix}^c = v_i^c \cos\theta_i^c, \dot{y}_{ix}^c = v_i^c \sin\theta_i^c, \dot{\theta}_i^c = u_i \tag{6}$$

where $u_i \in [-c, c]$ is chosen to minimize the deviation from w_i and to ensure that the trajectory has the same path length as w_i. Note that if $u_i = +c$, then the DTS given in Eq. (6) traces out a right-handed circle, as shown in Fig.3. Similarly, if $u_i = -c$, then the DTS traces out a left-handed circle. As shown in Fig.3.

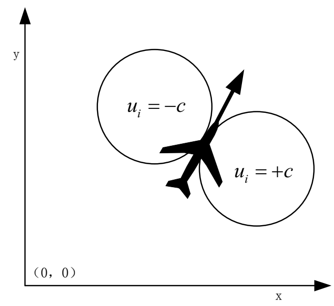

Fig.3 Control Strategies of the DTS.

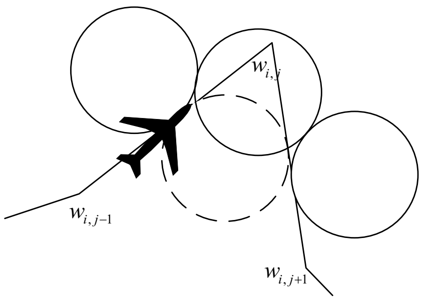

Fig.4 Length-matching transitions between way-point path segments

Consider the problem of turning from a path onto another in minimum time at constant velocity. Because the Voronoi path is used only for the initial path, we should have the smoothed trajectory which have the same length as the original Voronoi path. As Fig. 4 shows, our approach is offering the transition circle between the inscribed circle and the circle that intersects the waypoint, a transitioning trajectory can be determined that has the same length as the original Voronoi path.

3. Strategies for a simultaneous attack based on DPSO

3.1. Strategies for simultaneous attacks

For simultaneous attacks on each target, a timing constraint is needed. Once grouping has been completed and a path from each UAV to each target has been selected, the required time for arrival can be estimated. After the target is assigned and the path selected, each UAV's time over target (TOT) [0] can be computed, and we can select best TOT for each team. Because this assignment and path planning are at the highest level of mission operation, the TOT can be simply calculated as Eqs. (9-10):

$$\min T_{k,i} = L_{k,i} / V_{k,\max} \tag{9}$$

$$\max T_{k,i} = L_{k,i} / V_{k,\min} \tag{10}$$

Where the subscript k denotes the identification number of UAV(k),and i denotes the target(i). So $L_{k,i}$ is the length of the selected path of UAV(k) assigned to target(i). m is the mumber of uav teams. In case of rendezvous, the best TOT of each team, T* can be determined by Eqs. (11-13):

$$T_i^* = \min t, t \in T_i \tag{11}$$

$$T_i = T_{1,i} \cap T_{2,i} \cap \cdots \cap T_{m,i} \tag{12}$$

$$T_{k,i} = \{t \mid \min T_{k,i} \le t \le \max T_{k,i}\} \tag{13}$$

Therefore, the timing constraint can be given as though the T* determined by Eq.(13) must exist.

3.2. DPSO Algorithm

PSO is a population based stochastic optimization technique developed by Eberhart and Kennedy..Each particle in a swarm is a potential solution in the search space and particle adjusts its velocity according to its own flying experiences and its flock's experiences. The PSO technique is similar to the evolutionary computation techniques, however, in PSO each particle can adapt its velocity to move in the search space and has memory of its best position.

In order to solve the discrete UAV task assignment, we should use the Discrete PSO(DPSO) instead of basic PSO algorithm. By assuming the optimization problem to be of Q-dimension, then each particle in the swarm S can be represented as Eqs.(14-15) :

$$V_{id} = w \otimes V_{id} \oplus r_1 \otimes (p_{id} - x_{id}) \oplus r_2 \otimes (p_{gd} - x_{id}) \tag{14}$$

$$x_{id} = x_{id} + V_{id} \tag{15}$$

$x_i = (x_{i1}, x_{i2}, \cdots x_{iQ})$, $i = 1, \cdots, |S|$ .The best previous position attained by the ith particle is represented as $P_i = \{p_{i1}, p_{i2}, \cdots, p_{iD}\}$, and the best position in swarm is represented as $P_g = \{p_{g1}, p_{g2}, \cdots, p_{gD}\}$. $\otimes$, $\oplus$ means the velocity multiplication intercepting operation and the velocity additive operation respectively. The evaluate function is given as:

$$J(X_i) = \min(J(X_i)_1, J(X_i)_2, \cdots, J(X_i)_n) \tag{16}$$

$J(X_i)_j$ denotes the weight of the jth path form start location to target location through $x_{i1}, x_{i2}, \cdots, x_{im}$.

First the DPSO algorithm parameters are initialized in random integral patterns. For each iteration, all the particles are evaluated using evaluate swarm function. Before evaluating the particle in this function we check if weights of path exist or not. If they do not exist, then the value of the particle is set to infinity. Otherwise the particle is evaluated . If the particle does not have a solution then its value is set to infinity.

4. Simulation analysis

We consider a sample mission scenario with 4 UAVs and 2 targets and analyze the computation result taken by the algorithm and the time taken to accomplish the mission. The region is an area 30000m×30000m in size and there is 22 threat points, and the initial position of the UAVs and targets are given initially . For each UAV, the available speed range is given as $30m/s \le v_i \le 50m/s$.

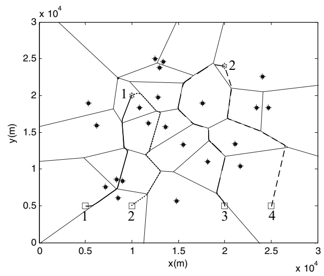

Fig.5 Simulation Results

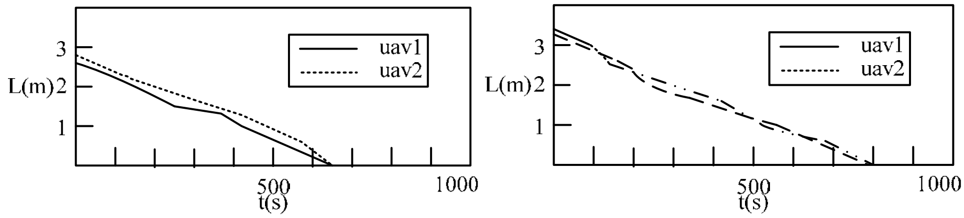

Fig. 6. (a) Distance from UAVs to assigned target in team1; (b) Distance from UAVs to assigned target in team2

Simulation results are presented for a team of 2 UAVs to each target. The objective is to avoid the threats while meeting the timing constraints imposed for the mission. In Fig.5, UAV1 and UAV2 attack target 1, UAV3 and UAV4 attack target 2, and all the paths are smoothed. In Fig.6, we can see that both teams attack targets simultaneously for each member.

5. Conclusions

In this paper we have outlined a cooperative control strategy based on Voronoi diagram and DPSO. Although the approach is sufficiently general to address a wide range of problems, we have applied it specifically to cooperative-trajectory planning problems involving timing constraints. Simultaneous arrival and movement constraints can each be accommodated using the cooperative-control algorithms and constraint formulations developed in this paper. the simulation results indicate that the proposed genetic algorithm can be used for the complicated group mission with timing constraints.

References

[1]P. B. Sujit, J. M. George and Randy Beard. Multiple UAV Task Allocation using Particle Swarm Optimization, AIAA Guidance, Navigation and Control Conference and Exhibit, August 18-21, Honolulu, Hawaii , AIAA-2008-6837

[2]Arthur Richards, John Bellingham, Michael Tillerson, and Jonathan How. Coordination and Control of Multiple UAVs, AIAA Guidance, Navigation, and Control Conference and Exhibit 5-8 August 2002, Monterey, California, AIAA 2002-4588

[3]Derek R. Nelson,TimothyW. McLain,and RandalW. Beard, Experiments in Cooperative Timing for Miniature Air Vehicles. Journal of Aerospace Computing Information and Communication. Vol. 4, August 2007,956-967

[4]Jung Leng Foo, Jared Knutzon, Vijay Kalivarapu, James Oliver and EliotWiner. Path Planning of Unmanned Aerial Vehicles using B-Splines and Particle Swarm Optimization. Journal of Aerospace Computing Information and Communication.Vol.6, April 2009,271-290

[5]Sivakumar Rathinam, Raja Sengupta, and Swaroop Darbha. A Resource Allocation Algorithm for Multivehicle Systems With Nonholonomic Constraints.IEEE Transactions on Automation Science and Engineering, VOL. 4, NO. 1, JANUARY 2007. 98-104

[6]Da-Wei Gu, Waseem Kamal and Ian Postlethwaite. A UAV Waypoint Generator, AIAA 1st Intelligent Systems Technical Conference 2004,Chicago,Illinois AIAA 2004-6227

[7]Ruiyun Yu, Xingwei Wang, Sajal K. Das.A Voronoi Diagram Approach for Mobile Element Scheduling in Sparse Sensor Networks, 2008 Second International Conference on Future Generation Communication and Networking,62-67

[8]Beard, R. W., McLain, T. W., and Goodrich, M. Coordinated Target Assignment and Intercept for Unmanned Air Vehicles, IEEE International Conference on Robotics and Autonomous, Inst. Of Electrical and Electronics Engineers, New York, May 2002.