

Multi-UAV Path Planning in Obstacle Rich Environments Using Rapidly-exploring Random Trees

Mangal Kothari, Ian Postlethwaite and Da-Wei Gu

Abstract—This paper presents path planning algorithms using Rapidly-exploring Random Trees (RRTs) to generate paths for multiple unmanned air vehicles (UAVs) in real time, from given starting locations to goal locations in the presence of static, pop-up and dynamic obstacles. Generating non-conflicting paths in obstacle rich environments for a group of UAVs within a given short time window is a challenging task. The difficulty further increases because the turn radius constraints of the UAVs have to be comparable with the corridors where they intend to fly. Hence we first generate a path quickly using RRT by taking the kinematic constraints of the UAVs into account. Then in order to generate a low cost path we develop an anytime algorithm that yields paths whose quality improves as flight proceeds. When the UAV detects a dynamic obstacle, the path planner avoids it based on a set of criteria. In order to track generated paths, a guidance law based on pursuit and line-of-sight is developed. Simulation studies are carried out to show the performance of the proposed algorithm.

I. INTRODUCTION

Recently, unmanned aerial vehicles (UAVs) have been deployed for various search and surveillance missions. One of the main objectives of a mission is to generate non-conflicting paths such that UAVs can follow them. A path is usually described in terms of a set of waypoints that the UAV has to visit sequentially. The path planning problem is to generate non-conflicting paths in terms of waypoints taking various environmental and physical constraints of the UAV into account. Designing a path planner for multiple UAVs in an obstacle rich environment that contains static, pop-up and dynamical obstacles is challenging. The difficulty further increases when a near optimal solution is sought in real time while accounting for kinematic constraints of the UAV.

For UAV applications, where the UAV has to fly in obstacle rich environments, the computational complexity of the path planning algorithm is an important issue that needs to be addressed. Since, the UAV is in motion at high velocity, the path planner has to find a path quickly. When there is a change in environment then they have to generate new paths in real time from their current locations to accommodate changes. If the path planner fails to generate a feasible solution within a given time window, a UAV may collide with the obstacles leading to mission failure.

The path planning problem is an important area of interest in the field of UAVs and robotics. A number of path planning algorithms have been developed in these fields such as road

map, cell decomposition, potential field, etc [1]-[2]. There are a few requirements for an ideal path planner, including optimality, completeness and computational complexity. However, few of these try to solve the planning problem in its full generality [1]. Depending on the nature of the problem, some techniques work better than others. An extensive introduction to the path planning problem and existing solutions may be found in [1]-[3]. The computational time of deterministic and complete algorithms grows exponentially with the dimension of the configuration space. Therefore these algorithms usually cannot be employed for real time UAV path planning problems, specially for the problems that contain rich static and pop-up obstacles. Recently, sampling based motion planning has gained much interest as an alternative to complete motion planning algorithms [4]-[5]. For example, Rapidly-exploring Random Trees (RRTs) have been shown to be effective in UAV applications [6]-[7].

In this paper, we develop a path planning algorithm for multiple UAVs traversing from their starting locations to corresponding goal locations in the presence of both static and dynamic obstacles. When a UAV detects a new obstacle, a new path is first found quickly and then depending on the time available an anytime algorithm is run to produce a better path. In order to track the generated path closely, a robust guidance law based on pursuit and line-of-sight is developed.

The rest of paper is organized as follows. In Section II we describe a scenario for the mission and the problem statement. In section III, we state the algorithm for path planning for multiple agents. In Section IV, in order to track a generated path, a guidance law is developed. In section V, the performance of path planner is shown by simulation results and finally the paper is concluded in section VI.

II. PROBLEM FORMULATION

A. Scenario

A mission is considered where multiple UAVs need to travel from their starting locations to goal locations through an environment consisting of static, pop-up and dynamic obstacles.

We assume full knowledge of the terrain map, however, pop-up obstacles may be encountered during flight. The problem becomes difficult because the motion constraints of UAVs (e.g. turn radius) are comparable with the environment. This necessitates that UAV constraints must be taken into account while finding a path so that unexpected collisions can be avoided. The objective here is to determine a path which is near optimal in real time while avoiding any known

M. Kothari, a graduated student, D.-W. Gu and I. Postlethwaite, Professors, are with Control Research Group, Department of Engineering, University of Leicester, United Kingdom, LE1 7RH. mk221@le.ac.uk, dag@le.ac.uk, ixp@le.ac.uk

a priori, pop-up and dynamic obstacles while satisfying physical constraints.

B. Assumptions and constraints

It is assumed that the UAVs are subjected to limited sensor and communication ranges, constant velocity and physical constraints. Since the UAVs have limited sensor range, they can only detect pop-up obstacles when they are in the sensor range. If the obstacles are on the flight path, the UAV has to re-plan its path from its current position. Depending on the velocity of the UAV, it can predict the time to collide T_i^c and has to find a new path before T_i^c . The UAV cannot make an instant turn, and hence, it has to start a collision avoidance manoeuvre earlier than the predicted time of collision T_i^c , which is denoted as T_i^m (time to manoeuvre). Determining a path which satisfies physical constraints while avoiding pop-up obstacles in the obstacle rich environment is difficult and computationally intensive. The UAV may take a detour route to avoid a collision, however, to find such a detour route which satisfies turn radius constraints and leads to the goal location is not trivial. Hence, we develop an algorithm combining RRT and anytime approaches to generate a time dependent improvised path. The UAVs are also subjected to limited communication range, and hence, they cannot inform other UAVs about their planned path that may be in conflict. In this case, when UAVs arrive in communication range of each other they exchange their route and determine the time of conflict T_i^{con} and time of manoeuvre T_i^m . If there exist, a new path is determined. Designing a path planner with these constraints is difficult and is the main contribution of this paper.

III. RRT BASED SOLUTION

In this section, we shall first discuss the standard RRT algorithm and its variant. Then, we develop a strategy to convert differential constraints to geometrical constraints in order to incorporate actual trajectory characteristics into the path planner. Then, we use a greedy strategy which not only enhances the computational capability but also reduces unnecessary path segments. Combining the greedy strategy with an anytime approach, a suboptimal path planner is developed which avoids static and pop-up obstacles. Finally, using this path planner a Multi-UAV path planner is developed.

A. Standard RRT algorithm

The objective is to find a path to travel from a starting position (x_{start}) to a goal position (x_{goal}) through a configuration space. Since the early development of mission planning algorithms, a large number of algorithms have been developed for such a problem. To name a few, one can find, road map, cell decomposition, potential field, visibility line [1],[8],[9]-[11]. However, solving the path planning problem while accounting for geometric and differential constraints is really challenging. Lavelle [6] developed the Rapidly-exploring Random Tree (RRT) method as a potential solution for such problems. The standard RRT algorithm is given in Algorithm 1.

Algorithm 1 Standard RRT Algorithm

- 1: Choose an initial node x_{init} and add to the tree τ
- 2: Pick a random state x_{rand} in the configuration space C
- 3: Using a metric ρ , determine the node x_{near} in the tree that is nearest to x_{rand}
- 4: Apply a feasible control input u to move the branch from x_{near} towards x_{rand} for a pre-chosen incremental distance
- 5: If there is no collision along this branch, add this new node x_{extend} to the tree τ
- 6: Repeat steps 2 to 5 until x_{goal} is included in the tree τ and find the complete path from x_{init} to x_{goal}

B. Path planner for single UAV

1) *Greedy strategy:* As already mentioned, because of random incremental growth of the RRT, the resulting path will not be straight for long distances. For straight paths in long corridors, a greedy strategy is proposed, which exploits the basic incremental growth characteristic while preserving random features. In this strategy, the incremental growth is continued until an obstacle is encountered, which is different from the modified RRT algorithm. This strategy will curb the tendency of growing too many path segments and will speed up the algorithm.

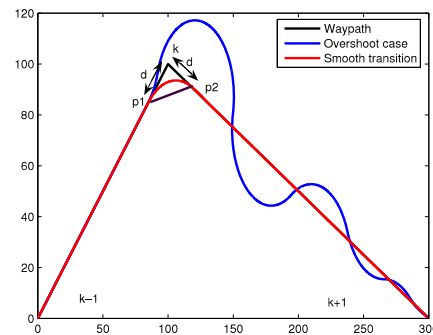


Fig. 1. Smooth switching

2) *Incorporation of actual trajectory characteristics:* The true UAV trajectory usually deviates from the waypoint path [12], specially near the edges. These deviations can be predicted for the given kinematic model, and hence, collisions with obstacles can be avoided. With this information, the path planning algorithm must ensure that the actual trajectory is also free from collisions.

To address this problem, we have developed a mechanism which enables the UAV to transit smoothly. If the UAV traverses to the waypoint k as shown in the Figure 1 before switching to waypoint $k+1$, it will overshoot and may collide with obstacles, which is undesirable. It can be observed from the figure that these oscillations are significant for the sharp turn, which results in further danger to the mission. For the success of the mission, one has to predict the overshoots and check for collisions. The actual trajectory of the UAV can be parameterized in terms of straight lines and circular

arcs. Since the new node (which is selected randomly) is known, we use this information to parameterize the actual trajectory of the UAV in the terms of straight lines and the arcs which satisfy the minimum turn radius constraint. The deviation due to the kinematic constraint now stays within the isosceles triangle $\triangle p_1kp_2$ (figure 1), and the path planner has to ensure that there should be no obstacles in it. In this way, we have successfully converted differential constraints into geometric constraints with an additional obstacle for each new node. The UAV starts taking the turn at the point p_1 and continues in the turn until it arrives at point p_2 as shown in Figure 1. With a good path following technique, the UAV should be able to track the trajectory as long as the radius of arc is greater than the minimum turn radius R_{min} . The steps of the proposed approach incorporating actual trajectory characteristics are given in Algorithm 2.

Algorithm 2 RRT Algorithm with incorporation of greedy strategy and actual trajectory characteristics

- 1: Choose an initial node w_{init} and start the tree τ
 - 2: Pick a random waypoint w_{rand} with little bias
 - 3: Using a metric ρ , determine the node w_{near} in the tree that is nearest w_{rand}
 - 4: Extend the branch from w_{near} toward w_{rand} by an incremental distance, resulting in node w_{extend}
 - 5: If there is no collision along this branch
 - 6: **if** check the collision with actual trajectory **then**
 - 7: add node w_{extend} to the tree τ
 - 8: **else**
 - 9: go to step 2
 - 10: **end if**
 - 11: Keep extending the branch by an incremental distance and keep adding new nodes w_{new} to the tree τ until an obstacle is encountered
 - 12: Repeat steps 2 to 11 until w_{goal} is included in the tree τ
 - 13: Find the complete path from w_{init} to w_{goal}
 - 14: Eliminate extraneous nodes from this path while checking the turn radius constraint for actual trajectory
-

3) *Anytime approach*: As argued before, the anytime approach provides solutions for any given computational time and the solutions get better as the computation time increases. The idea is first to generate a path using Algorithm 2 for a given scenario. When the UAV flies from one waypoint to another waypoint, the processor stays idle when the path is being tracked. The anytime algorithm exploits this idle position state and uses the available computation power to find a new less costly path. This strategy is followed until the UAV reaches the target point. The steps of the proposed approach are shown below in Algorithm 3.

It is clear that the proposed algorithm offers as a bonus a useful approach for avoiding pop-up obstacles. The algorithm has the capability to deal with disturbances as well. If the UAV deviates significantly from its original path so that returning to the path may endanger the mission, a new search is carried out from the vicinity of the UAV current flying area to find a new path.

Algorithm 3 RRT Algorithm for Suboptimal Solution

- 1: Find a path for given space using Algorithm 2
 - 2: **while** Until UAV reaches the target point **do**
 - 3: Search for low cost paths using Algorithm 2 from next waypoint while UAV tracking a line connected immediate and next waypoints
 - 4: **if** An obstacle is encountered in waypoints path and in the sensor range **then**
 - 5: Search for a path from vicinity of UAV position to the goal location
 - 6: **else**
 - 7: Keep tracking until UAV reached intended waypoint
 - 8: **end if**
 - 9: Select new waypoint based on low cost path
 - 10: **end while**
-

C. Path planner for multiple UAVs

Generating non-conflicting paths for several UAVs that are flying simultaneously is a challenging task due to the space constraints that one path imposes on the other and the physical constraints which are specially important in an obstacle rich environment. In this work, we have considered a path planner for each UAV. Each UAV has its own path planner that exchanges some information with the others when they are in communication range. This information is needed to resolve the conflict while accounting for the mission objectives.

Coordination objective(s)

At the beginning of the mission, it is assumed that the number of obstacles, the location of the obstacles and their shapes are known. With this information UAVs can plan their individual paths. If the start locations of the UAVs are within the communication range then they can share their plans. If paths are in conflict that can result in a collision then the UAVs need to coordinate with each other to change their paths to avoid collision. Let T_i^{con} be the minimum time to be allowed to compute a new path before a collision can occur. In that case, we run Algorithm 2 for time T_i^{con} with the updated world having an additional obstacle at the point of conflict. The algorithm keeps running until the time when it returns the best available path. Note that in the case of the conflict(s) not all UAVs involved need to search for new paths. Based on coordinative object just a UAV is assigned to find a new path. In this work, we have considered the following criteria for coordination; (i) the amount of fuel left; (ii) the distribution of the load and (iii) token number i.e. priority of the task of involved vehicle (if that is known). The implementation of this is discussed in the simulation results. The steps of the proposed approach for multiple UAVs are shown below in Algorithm 4.

IV. UAV DYNAMICS AND GUIDANCE LAW

Each UAV has to follow the path generated by the proposed approach. In order to track a given path, a guidance law is required which allows the UAV to track the path with minimum deviation. For this purpose, we employ a guidance law that is a combination of a pursuit guidance

Algorithm 4 Path Planner for multiple UAVs

```

1: Find a path for UAV  $u_i$  using Algorithm 2,  $\forall k = 1, \dots, N$ 
2: for  $k = 1$  to  $N$  do
3:   while Until all UAV reach to the their corresponding
      target points do
4:     Search for low cost paths using Algorithm 2 for
      UAV  $u_i$  from next waypoint while tracking
5:     if A pop-up obstacle is encountered in waypoints
      path then
6:       Search for a new path from vicinity of UAV
      position
7:     else
8:       Keep tracking until UAV reached next waypoint
9:     end if
10:    Select new waypoint based on low cost path
11:    if UAV  $u_i$  is in the communication range of the
      UAV  $u_j, j \neq i$  then
12:      Find whether paths are in the conflict
13:      if paths are in conflict then
14:        Determine a new path from vicinity of UAV
        (either  $u_i$  or  $u_j$ ) position based on certain
        criteria
15:      end if
16:    end if
17:  end while
18: end for

```

law and a line-of-sight (LOS) guidance law taken from the missile guidance literature. This choice is motivated from the fact that pursuit allows the UAV to reach the desired waypoint quickly and it is simple to implement. However, the drawback is that its trajectory may not be along the LOS as desired. Hence, an LOS guidance law that will steer the UAV towards the LOS is needed to overcome this difficulty. To design a guidance law, we need a navigation model of the UAV. For the purpose of illustration, we consider the following three state navigation model of a UAV

$$\dot{x} = V \cos \alpha \quad (1)$$

$$\dot{y} = V \sin \alpha \quad (2)$$

$$\dot{\alpha} = \frac{a}{V} \quad (3)$$

where V represents velocity, α represents the heading angle and a represents lateral acceleration (latax) which acts as a guidance parameter. The bounds on a are as follows $-a_{max} \leq a \leq a_{max}$.

Consider the path following geometry of the UAV as shown in Figure 2. Assume that the UAV was initially located at the x_{w1}, y_{w1} and it has to follow the path formed by the waypoints W_i and W_{i+1} from its current position x, y . For the UAV to move towards the waypoint W_{i+1} , it has to change its course so it aligns itself along the line connecting the waypoints. The required latax to align heading along LOS is calculated as follows

$$a = k_1 (\alpha_d - \alpha) + k_2 R_m \sin(\theta_T - \theta_M) \quad (4)$$

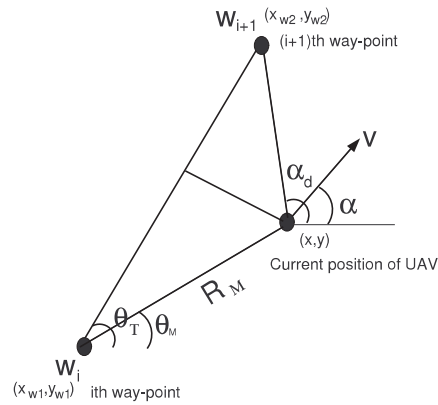


Fig. 2. UAV guidance geometry

Here k_1 and k_2 are design parameters and other terms are defined as $\alpha_d \triangleq \tan^{-1} \left(\frac{y_{w2} - y}{x_{w2} - x} \right)$, $\theta_T \triangleq \tan^{-1} \left(\frac{y_{w2} - y_{w1}}{x_{w2} - x_{w1}} \right)$ and $\theta_M \triangleq \tan^{-1} \left(\frac{y - y_{w1}}{x - x_{w1}} \right)$. where x_{w1}, y_{w1} and x_{w2}, y_{w2} are the x and y coordinates of the waypoints W_i and W_{i+1} respectively. The first term in (4) represents a pursuit policy which guides the UAV towards the way-point whereas the second term in (4) represents the LOS policy which tries to minimize separation from the LOS. As mentioned in Section III-B.2, in order to traverse smoothly to the next waypoint, the UAV has to start turning before it arrives to the next waypoint. Consider the waypath geometry as shown in Figure 1. Assume that the UAV is approaching towards waypoint k and it has to move to waypoint $k + 1$. It has to start to turn at point p_1 and follow the arc until it reaches point p_2 . The distance d of points p_1 and p_2 from waypoint k is a function of arc radius and the angle between the waypath as follows $d = \frac{R}{\tan \beta}$ where 2β is the angle between waypath formed by joining waypoint $k - 1$ to k and k to $k + 1$. The guidance law to follow the arc is different from that to follow a straight line, since the LOS term is not required to follow the arc. The distance d provides the measure for switching from a pursuit law to combined pursuit and LOS laws and vice versa.

Performance of the guidance law

We have developed a guidance law with a switching mechanism which enables the UAV to follow the waypath smoothly. The switching mechanism uses a measure which is a function of turn radius and angle between consecutive waypaths. Figure 3 depicts the performance of the guidance law in which a UAV has to travel the path as shown in Figure 3. It can be observed from the figure that that UAV starts turning before it reaches the waypoint to traverse the next waypath smoothly. After it completes the turn, it follows the desired straight line with minimum deviations.

V. SIMULATION RESULTS

In this section, we present simulation studies to demonstrate the performance of the proposed algorithm. The objective is that a set of UAVs has to travel from their given starting locations to goal locations while avoiding static, pop-up and dynamic obstacles in an obstacle rich environment. It

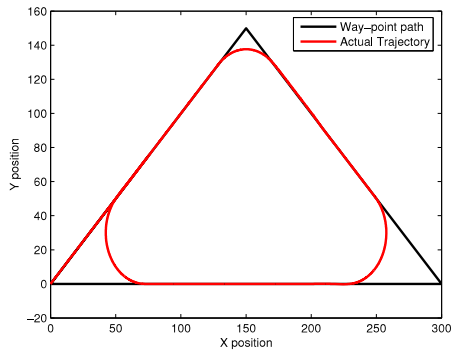


Fig. 3. Performance of the guidance law

is assumed that all UAVs are flying at the same height. Now we will demonstrate the effectiveness of the path planning algorithm based on the few example scenarios.

A. Path planning in obstacle rich environment for multiple UAVs

As discussed before, the algorithm has the capability to provide collision free paths while satisfying a turn radius constraint. We have tested the performance of the path planner with several scenarios. The position and size of obstacles are chosen randomly, however, lengths and widths of obstacles are kept above the prefixed minimum length and width. Each time the path planner finds the flyable path quickly. In Figure 4, we find paths in a terrain of $5 \times 5 \text{ km}^2$ with 1500 obstacles. It can be observed from the figure that the paths are free from collision. The tracking performance of the path planner is checked by deploying the guidance law developed in Section IV. Planned and executed trajectories are almost on top of each other. The closeness of the trajectories verify the benefits of incorporating actual trajectory characteristics into the path planning algorithm. Since every UAV is mounted with its own path planner, the computation complexity to find a path will not increase with the number of UAVs in the mission.

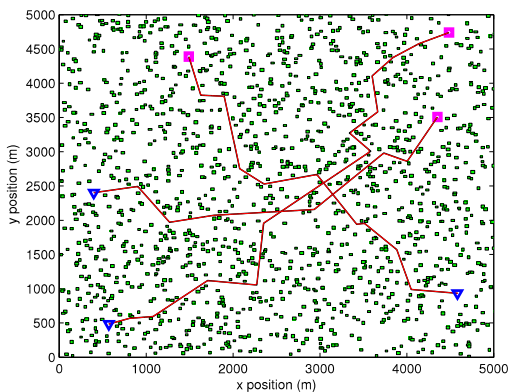


Fig. 4. Performance of path planner in obstacle rich environment

B. Path planning with pop-up obstacles for multi-UAV

Obstacles are classified into two types: a priori known obstacles from which a trajectory can be generated before flight, and pop-up obstacles encountered during flight. A terrain map is usually used to generate initial waypoints, and a local path planning scheme such as bouncing [8] is used to avoid pop-up obstacles. Here we deal with both types of obstacles through the proposed algorithm because it is able to provide solutions very quickly. It is assumed that the UAVs are mounted with onboard sensors which can detect obstacles if they are on a collision course. As soon as the UAVs detect an obstacle, the algorithm incorporates this information into the terrain map and starts searching for a new path. Figure 5 shows such a scenario in which pop-up obstacles are encountered in the waypath. The pop-up obstacles are shown in red blocks in the figure whereas green blocks represent static obstacles. The blue lines show the new path after pop-up and the red line represents the actual trajectory of the UAV. It can also be noted from the figure that other UAVs do not replan their paths because they do not have any information about pop-up obstacles.

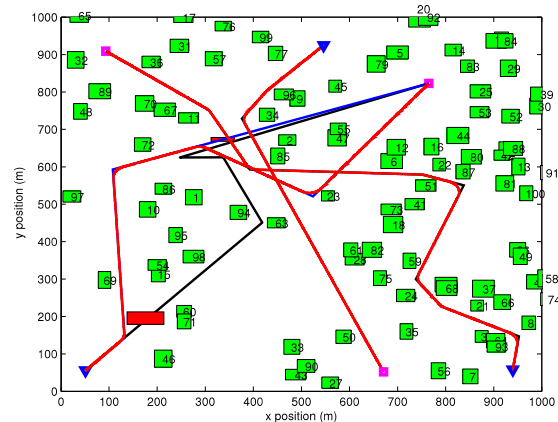


Fig. 5. Path planning with pop-up obstacles

C. Path planning with anytime approach for multi-UAV

There are several considerations for an ideal path planner including optimality, completeness and computational complexity. In general the optimal solution demands high computation, and hence cannot be computed in real time. The RRT algorithm generates a quick solution but it is far from optimal due to its random exploration of the space. The idea here is to generate suboptimal solutions while preserving quickness of the algorithm. This is achieved with the anytime approach to develop a suboptimal path planner which not only provides a solution quickly but the quality of solution gets better as the computation time increases. The anytime approach uses the idle state of the processor, when the UAV flies between two given waypoints, to find new paths. The search is carried out during the time it flies from one waypoint to another. If the cost of any new path is less than the previous path it is accepted; otherwise the

UAV follows the earlier path. This search is continued until the UAV reaches the target. Figure 6 shows the trajectories with the anytime approach strategy. It can be observed that during the flight the proposed algorithm always tries to find a lower cost path to the next waypoint.

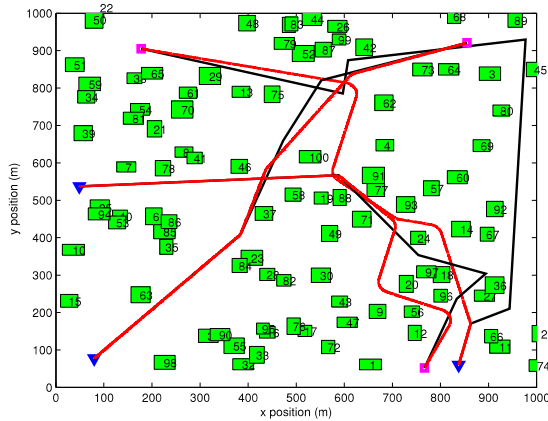


Fig. 6. Path planning with anytime approach

D. Path planning with dynamic obstacles

As argued, if paths are in conflict that can result in collisions, then the UAVs need to coordinate with each other to change their paths to avoid collision. The coordinated objective should be decided such that the mission completes within the shortest possible time and with the minimum resources while resolving conflict. We have considered the following criteria to resolve conflict: (i) the amount of fuel left in a UAV and the distance to be covered by it; (ii) the amount of resources to be needed to complete the mission and try to distribute load equally (the load is equally disturbed by asking to travel almost equal distance) and (iii) token number. It means the UAV has higher token number will determine the path and the UAV has lower token number will continue with the planned path. Note that we have considered a bubble as conflict (intersecting region) instead of a point in time and space, so UAVs need not to travel very close to each other. Figure 7 shows a case in which two UAVs are in conflict. It can be seen that the first UAV continues its path whereas the second UAV change the path (since it has to travel less to reach its goal point) to avoid the collision, hence dynamic collision is avoided.

VI. CONCLUSIONS

In this paper, a path planning strategy for multiple UAVs in the obstacle rich environments is proposed. The strategy employs a path planner for each UAV and UAVs share information when they are in communication range. The algorithm performs well and provides collision free paths in the presence of static, pop-up and dynamic obstacles. The algorithm finds a path in a reasonably short time and the quality of solutions improve as computational time increases. In the case of conflict (dynamic collision), the path planner

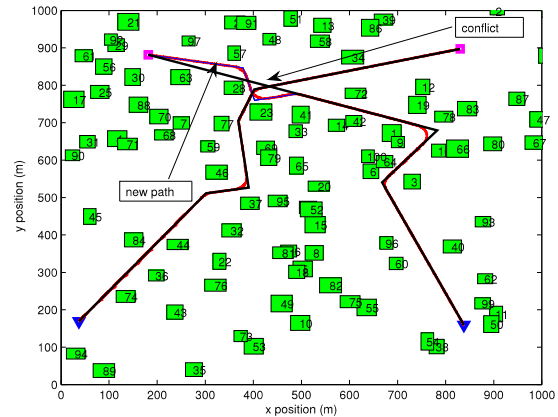


Fig. 7. Multi-UAVs path planning in conflict

find a new path based on a set of criteria to resolve the conflict. The proposed guidance law keeps the vehicle on the waypoint most of time. The performance of the path planner is verified by various simulation studies and is found to be satisfactory.

REFERENCES

- [1] J. Latombe, "Robot Motion Planning," Boston: Kluwer Academic Publishers, 1991.
- [2] S. M. LaValle, "Planning Algorithms," Cambridge University Press, 2006.
- [3] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, "Principles of Robot Motion: Theory, Algorithms, and Implementations", Cambridge, MA: MIT Press, 2005.
- [4] N. M. Amato and Y. Wu, "A Randomized Roadmap Method for Path and Manipulation Planning," in Proceedings of the IEEE International Conference on Robotics and Automation, 1996, pp. 113-120.
- [5] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces". IEEE Transactions on Robotics and Automation, Vol. 12, 1996, pp. 566-580.
- [6] S. M. LaValle, "Rapidly-exploring Random Trees: A New Tool for Path Planning," 1998, TR 98-11, Computer Science Dept., Iowa State University.
- [7] J. B. Saunders, B. Call, A. Curtis, R. W. Beard, and T. W. McLain, "Static and Dynamic Obstacle Avoidance in Miniature Air Vehicles", AIAA Guidance, Navigation, and Control Conference and Exhibit, 2006.
- [8] D. W. Gu, I. Postlethwaite and Y. Kim, "A Comprehensive Study on Flight Path Selection Algorithms", Dept. Of Engineering, University of Leicester.
- [9] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," in Numerische Mathematik. Springer, Vol. 1, 1959.
- [10] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybernetics, Vol. 4, No. 2, 1968.
- [11] J. J. Kuffner and S. M. LaValle, "RRT-Connect: An Efficient Approach to Single-query Path Planning," in Proceedings of the IEEE International Conference on Robotics and Automation, 2000, pp. 995-1001.
- [12] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time Motion Planning for Agile Autonomous Vehicles," AIAA Journal of Guidance, Control and Dynamics, Vol. 25, No. 1, 2002, pp. 116-129.
- [13] M. Kothari, D.-W. Gu, and I. Postlethwaite, "An Intelligent Sub-optimal Path Planning Algorithm Using Rapidly-exploring Random Trees", submitted to European Control Conference 2009
- [14] D. Ferguson and A. Stentz, "Anytime, Dynamic Planning in High-dimensional Search Space", in Proceedings of the IEEE International Conference on Robotics and Automation, 2007, pp. 1310-1315.