J Intell Robot Syst (2013) 70:303–313
DOI 10.1007/s10846-012-9740-3

An Integral Framework of Task Assignment and Path Planning for Multiple Unmanned Aerial Vehicles in Dynamic Environments

Sangwoo Moon · Eunmi Oh · David Hyunchul Shim

Received: 29 June 2012 / Accepted: 13 July 2012 / Published online: 12 September 2012

Abstract In this paper, a hierarchical framework for task assignment and path planning of multiple unmanned aerial vehicles (UAVs) in a dynamic environment is presented. For multi-agent scenarios in dynamic environments, a candidate algorithm should be able to replan for a new path to perform the updated tasks without any collision with obstacles or other agents during the mission. In this paper, we propose an intersection-based algorithm for path generation and a negotiation-based algorithm for task assignment since these algorithms are able to generate admissible paths at a smaller computing cost. The path planning algorithm is also augmented with a potential field-based trajectory replanner, which solves for a detouring trajectory around other agents or pop-up obstacles. For validation, test scenarios for multiple UAVs to perform cooperative missions in dynamic environments are considered. The proposed algorithms are implemented on a fixed-wing UAVs testbed in outdoor environment and showed satisfactory performance to accomplish the mission in the presence of static and pop-up obstacles and other agents.

Keywords Task assignment · Path planning · Multi-UAV coordination · Multiple UAVs · Flight experiment

This work was supported in part by the Korea National Research Foundation Grant 20110015377.

S. Moon
Department of Aerospace and Mechanical Engineering, Korea Air Force Academy, Cheongju, South Korea
e-mail: dear.sangwoo@gmail.com

E. Oh
CNS/ATM & Satellite Navigation Research Center, Korea Aerospace Research Institute, Daejeon, South Korea
e-mail: emoh@kari.re.kr

D. H. Shim (✉)
Department of Aerospace Engineering, KAIST, Daejeon, South Korea
e-mail: hcshim@kaist.ac.kr

1 Introduction

UAVs are nowadays deployed for various missions including surveillance and security enforcement. As UAVs are deployed in more complicated missions, they may fly in areas filled with obstacles, which may not be known *a priori*. Also, when a mission becomes more complicated, it is desirable to "share the burden" by multiple UAVs. In such scenarios, UAVs should be able to fly through the desired waypoints without crashing into obstacles or other agents. Therefore, task assignment and path planning are two essential elements of mission planning and execution for

multiple UAVs. Mixed Integer Linear Programming (MILP) is a powerful method for task assignment because it can handle dynamic systems with discrete decision variables and there are many efficient solvers available. Bellingham used MILP for task assignment to handle waypoint visiting problems [1]. However, the complexity of the problem rapidly grows as the number of variables increases and therefore it is often not suitable for real-time applications. Furthermore, since the MILP is formulated with object functions for the entire system, it needs to be implemented as a centralized system, which may not function well when the central control station or communication links are broken.

Path planning is an important requirement for collision-free operation for UAVs. A path planner should satisfy completeness, optimality, computational tractability, and scalability. In addition, the collision and threat avoidance is required for multi-agent scenarios and there are varieties of algorithms for this purpose. Shim applied model predictive control (MPC) algorithm to a collision avoidance problem of sixteen homogeneous rotary UAVs [2, 3]. Schouwenaars et al. used mixed integer programming for multi-vehicle path planning [4]. Richards showed an aircraft trajectory planning with collision avoidance using MILP [5].

Also, some variations of MILP have been applied to this problem. Earl and D'Andrea developed an iterative MILP approach [6] and Jain and Gossmann proposed hybrid MILP/CP (Constrained Programming) models for a class of the optimization problems [7]. Chang and Shadden introduced the gyroscopic forces for collision avoidance, which is similar to the potential field approach [8].

In this paper, we propose a hierarchical framework for efficient path planning and task assignment for multiple UAVs in dynamic environments. The proposed path planning algorithm is based on the shortest-path principle in Euclidean space, which is combined with A^* search algorithm. Since the path planner is not able to handle any pop-up obstacles or other agents in real time, it is augmented with a potential field-based collision avoidance layer, which is responsible for solving a detouring trajectory around the detected obstacles.

For task assignment, we propose a negotiation-based algorithm, which searches for a solution by negotiating for lower costs to perform the given task at individual agent level. Initially, each agent chooses their first task depending on the cost, which is directly proportional to the distance to the task. The choices of all other agents are notified to each agent and, when there are conflicts, the costs of conflicting agents are compared, and the agent with the lowest cost has the priority to claim the task while the remaining agents repeat the same process until all conflicts are resolved. The proposed algorithms are computationally light enough to handle real-time path planning in partially known environment.

The proposed algorithms are first validated in simulations, where multiple UAVs are required to perform the given tasks while avoiding obstacles and other agents. Then, we perform a flight test using the same algorithms implemented for real world operation with multiple fixed-wing airplanes in outdoor environment with no-fly zones and pop-up threats. The proposed algorithms were shown to be capable of real-time path planning and task assignment in outdoor environment even with simulated dynamic obstacles.

2 Problem Formulation

Tasks can be defined as a set of actions to accomplish a job, a problem, or an assignment. It can be represented as a number of lower-level goals that are necessary for achieving the overall goal of the system [9, 10]. In this paper, each task is defined as visiting a waypoint, through which at least one agent should pass once. First of all, we need to establish the rules for task assignment and path planning [11]:

(a) Obstacles are defined as the objects that obstruct the motion. In our research, they are represented as polygons, which can model real-world obstacles more accurately than circles or ellipses.
(b) The environment is modeled as a two-dimensional Euclidean space.
(c) UAVs must avoid all obstacles.

(d) All UAVs must remain in admissible areas, i.e., clear from the obstacles with a margin. Therefore, paths of all UAVs solved by the proposed procedure should be admissible. Paths can intersect, but should not lead to collision among UAVs.
(e) The task assignment and the path planning should run in real time.

In the following section, we will present the task assignment and the path planning algorithms based on rules listed above.

3 Negotiation-Based Task Assignment

During the last decade, many auction and negotiation algorithms have been developed for task assignment problems [12–15]. The task assignment algorithm in this paper is based on the negotiation algorithm. The proposed method should generate a feasible solution within a reasonable time for real-time applications. In addition, this algorithm is an *event trigger-based process*, which means that it runs only when an agent sends a message to invoke this algorithm. This is a desirable attribute for decentralized systems running in real time.

Events can be classified into three cases: i) a mission is given to the system and the agents start to achieve the goal of a mission, ii) a task in a given mission is accomplished by an agent, and iii) a given mission is completely finished. If an event occurs in the mission, the presented task assignment process is activated. In that case, whole agents can be assigned with the different tasks, and this result is dependent on the conditions of tasks. On the other hand, the costs for negotiation are defined by using Eq. 1. This formula consists of two parts: the distance from the current location of an agent to the task location and the characteristics of an agent and the assigned task, i.e., the capability set of an agent and type of the task. The total cost is a linear combination of these two elements, and there are weights assigned for each term:

$$J_{task} = w_d J_{dist} + w_c J_{character} \tag{1}$$

In Eq. 1, J_{dist} is the cost related with the distance between the current position of an agent and the location of the given task. If there are tasks with different characteristics, $J_{character}$ can be given with different costs to affect the behaviors of all agents in this scenario. Herein this paper, without loss of generality, we assume the cost is dependent on the distance only so that $J_{character}$ in Eq. 1 is to zero.

The negotiation-based task assignment is performed in the following manner. Initially, each agent chooses their first task with the lowest cost, which is a function of the distance to the task. The choices of all other agents are notified to each agent and, when there are conflicts, i.e., more than one agents choose the same task, the costs of conflicting agents are compared, and the agent with the lowest cost has the priority to claim the task while the remaining agents repeat the same compare-and-claim process until all conflicts are resolved.

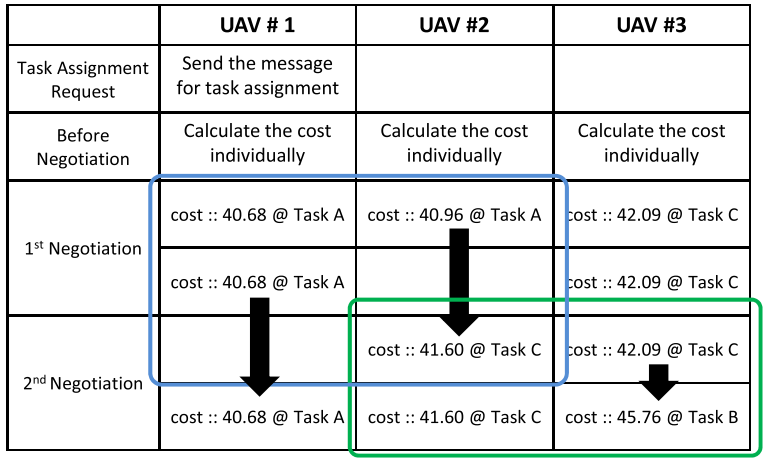

Fig. 1 Negotiation-based task allocation for three UAVs with three tasks

In Fig. 1, an example with three UAVs is given. At first, each UAV chooses their task and the cost is compared. At the first negotiation, UAV #1 and #2 choose the same Task A and they negotiate to find UAV #1 has lower cost. So UAV #2 is forced to choose a task other than A, and it chooses Task C with cost 41.60. This choice now conflicts with that of UAV #3 with cost 42.09, which is higher than the cost of UAV #2. Therefore, UAV #3 is forced to choose Task B with cost 45.76.

This approach has no guarantee to converge to the global minimum with an exception that it does converge to the global minimum when the number of the agent is equal to the number of tasks. However, this algorithm demands lower computing loads and suitable for decentralized scenarios with limited communication bandwidth.

4 Intersection-Based Path Planning

In the Euclidean space, the shortest path between two points is the straight line that connects these points. Denoting these points as $\mathbf{x}_s$ and $\mathbf{x}_g$, the cost function to find minimal distance—path can be written as

$$\min J = dist\left(\mathbf{x}_s, \mathbf{x}_g\right) = \left\|\mathbf{x}_s - \mathbf{x}_g\right\|. \tag{2}$$

However, if there are any obstacles or forbidden regions on the line, the shortest path is not admissible and a detouring path should be computed.

In order to find an admissible path in this situation, the intersection point by the path from Eq. 2 and the boundary of obstacles are used. The first step of the proposed algorithm is to find the intersection point. If an obstacle in the two-dimensional space is modeled as a polygon, then the intersection point $\mathbf{x}_m$ in Fig. 2 can be found easily. In such a case, a new path that passes the boundary points of a detected obstacle, $\mathbf{x}_{Oi1}$ or $\mathbf{x}_{Oi2}$ should be found. Therefore, these two points are the candidates of waypoints for path planning. The proposed algorithm uses A^* approach expressed as Eq. 3.

$$\begin{aligned} f(\mathbf{x}) &= g(\mathbf{x}) + h(\mathbf{x}) \\ &= \|\mathbf{x}_s - \mathbf{x}_w\| + \left\|\mathbf{x}_g - \mathbf{x}_w\right\| \end{aligned} \tag{3}$$

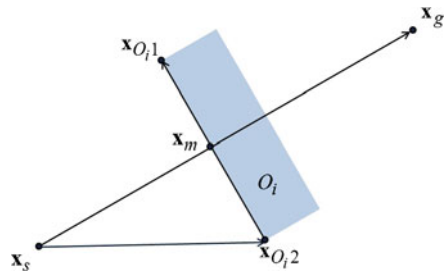

Fig. 2 Intersection point between shortest path and an obstacle

In Eq. 3, $\mathbf{x}_w$ is the boundary point of detected edge so that the first term of Eq. 3 is the distance from the starting point to a waypoint candidate and the other is the distance from the waypoint candidate to the goal point. If the detected line is not the first since the operation started, this equation should be modified to Eq. 4 because the vehicle is already assigned to a waypoint.

$$\begin{aligned} f(\mathbf{x}) &= g(\mathbf{x}) + h(\mathbf{x}) \\ &= \sum_{i=1}^{N} \left\|\mathbf{x}_{i\Delta t} - \mathbf{x}_{(i-1)\Delta t}\right\| + \left\|\mathbf{x}_g - \mathbf{x}_w\right\| \end{aligned} \tag{4}$$

In the two-dimensional space, the suboptimal waypoints and the candidate waypoints appear as a pair, and the suboptimal waypoints receive a higher priority for calculation. If multiple obstacles are located in the area, the intersection point which is the closest among the intersection points and the obstacle of that point is considered. Since A^* algorithm is used in this procedure, it is possible to find the globally optimal path using the nodes found using the intersection method described above. Therefore, the nodes related to

Table 1 Pseudo code for path planning

Algorithm 1 Intersection-based path planning

1: $P \leftarrow$ InitializePath(x_s, x_g)
2: $x_i \leftarrow x_s$
3: while dist(x_g, x_i) > $d_{admissible}$ do
4: $x_m \leftarrow$ DrawLine(x_g, x_i)
5: if x_m exist
6: $x_i \leftarrow$ SelectOptimalNode($x_g, x_s, x_1, x_2, \ldots, x_i$)
7: $P \leftarrow$ AddPath (P, x_i)
8: end if
9: $i \leftarrow i + 1$

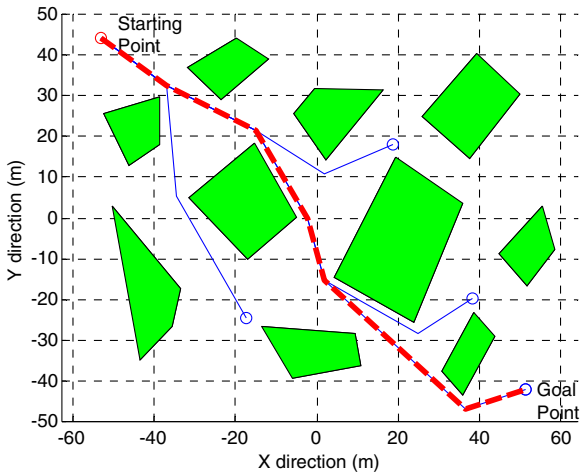

Fig. 3 Result from the proposed line-based oath planning. (*solid lines*: suboptimal paths selected as the optimal path candidates, *dashed line*: optimal path solved from the proposed algorithm)

the intersection points are used for the path planning, and the procedure from Eqs. 2 to 4 will be repeated until the solved node reaches the goal point (for detailed description on the algorithm, see Table 1).

Figure 3 shows a simulation result obtained using the proposed path planning algorithm. In this scenario, the area is filled with ten arbitrarily shaped obstacles. The dashed line is the solved path, and the solid lines are the candidate paths to compute the shortest path. First of all, the algorithm finds a path that passes on the left side of an obstacle. However, the priority is switched after this path meets another obstacle and a new candidate path is found. After several iterations, this algorithm determines that the path should be modified, and the final path is again replaced with the dashed line, which is the optimal path.

5 Potential Field-Based Collision Avoidance

The collision avoidance is a dynamic path generation technique, which finds an admissible path to avoid the collision among agents in the given environment. The algorithm proposed here is based on the potential field approach with some improvements explained below. Almost all algorithms related on the potential field are based on the distance between the vehicle and target points at one time frame. However, in case of moving threats or obstacles, it is desirable to consider the relative direction of motion as well. The proposed algorithm utilizes the cost function for the potential field as the function of the distance and direction of the obstacle using the normal vector as

$$J = fn\left(\mathbf{x}, \mathbf{v}, \mathbf{n}\right). \tag{5}$$

The main procedure of this algorithm is described as below. First of all, a set of path candidates over a finite horizon into the future are constructed. For each path, the cost is computed with respect to the attraction forces from the waypoints or goal points, and the repulsion forces from the agents. Among those candidates, the best candidate that has the lowest cost can be selected, and an UAV moves in one step along the selected path. This procedure is iterated until the UAV reaches to the target point.

The cost function for the waypoints generated from the path planning can be expressed as Eq. 6. Note that Eq. 6 is not dependent on the vehicle's heading angle. In order to prevent Eq. 6 from dominating other terms, a denominator that divides by the distance from the waypoint to the current position of an UAV is included.

$$J_{\text{waypoint}} = 1 - \exp\left(\frac{\left\|\mathbf{x}_{\text{waypoint}} - \mathbf{x}_{t+\Delta t}\right\|}{\left\|\mathbf{x}_{\text{waypoint}} - \mathbf{x}_{t}\right\|}\right) \tag{6}$$

The cost function for the obstacles and threats is expressed as Eq. 7, and again it is independent from the vehicle's heading. If the angle between the normal vector and velocity vector is close to zero, i.e., the vehicle approaches to the object, the cost function grows exponentially. However, if this angle is close to 180 degrees, the cost function decreases rapidly because the UAV moves apart from the object. In a similar reason explained above, a denominator is included to divide the term by a magnitude vector from the object to the present position of an UAV. In Eq. 7, $C_{obstacle}$ is a positive real number to introduce a adequate margin for any discrepancy between the actual obstacle and its polygon-based approximation and $\mathbf{n}_i$ is the normal vector which connects the edge of an obstacle and UAVs. Therefore, $\theta_{obstacle}$ in Eq. 7 is the angle between the normal vector $\mathbf{n}_i$

and velocity vector $\mathbf{v}_{t+\Delta t}$ after the finite time, and $\theta_{obstacle}$ affects the cost for obstacles.

$$J_{obstacle,i} = (0.5\cos\theta_{obstacle} + 0.5 + C_{obstacle}) \times \exp\left(-\frac{\|\mathbf{n}\|}{\|\mathbf{x}_t - \mathbf{x}_{obstable}\|}\right)$$

where

$$\theta_{obstacle} = \cos^{-1}\left(\frac{\mathbf{n}_i \cdot \mathbf{v}_{t+\Delta t}}{\|\mathbf{n}_i\|\,\|\mathbf{v}_{t+\Delta t}\|}\right) - \pi$$

$$\mathbf{n}_i = \left[\, E_x \;\; E_y \;\; E_z \,\right]_i^T - \mathbf{x}_{t+\Delta t} \tag{7}$$

6 Integrated Architecture for Multi-UAV Coordination

Figure 4 explains how the task assignment layer, path planner, and collision avoidance layer presented so far are integrated altogether. The integration of the negotiation-based task assignment and the intersection-based path planner is a crucial step for architecting a mission-planning framework for multi-agent scenarios. In our research, the task assignment layer needs to communicate with the path planner to receive the cost for task assignment, which is a function of distance to reach the destination. During the task assignment, the path planning layer is repeatedly called until a feasible solution for task assignment is found.

The next step is, if any unmapped obstacle is detected, to find a collision-free path using the original path computed by the path planner. It is noted that the path planner and the collision avoidance layer may appear to do a similar task of finding a conflict-free path. However, the collision avoidance layer finds a path locally detouring from the original path found by the path planner, which only uses the obstacle information known before starting the mission. Therefore, these two algorithms work in a complementary manner to generate the collision-free path efficiently.

As mentioned, these two procedures are in a hierarchical relationship and the path planning algorithm is invoked when the situation is in three cases: before carrying out the mission, a new task is allocated on an agent participating in the mission, and new obstacles or threats suddenly appear, or "pop up". The collision avoidance algorithm is activated when the agent go to the target point while following the path. For multi-UAV scenarios, the collision avoidance between two agents should be considered. The presented avoidance algorithm is then used to generate a collision-free path. In this result, the vehicle's kinematic constrains such as maximum turns rates or minimum turn radii. We assume the flight controller is capable of reaching the waypoints, which is the vertices of the generated path. In case the vehicle cannot reach the required waypoints promptly, the task assignment layer redistributes the task among the agents in the neighborhood. It is noted that our proposed method can be further improved by explicitly considering the vehicle's characteristics.

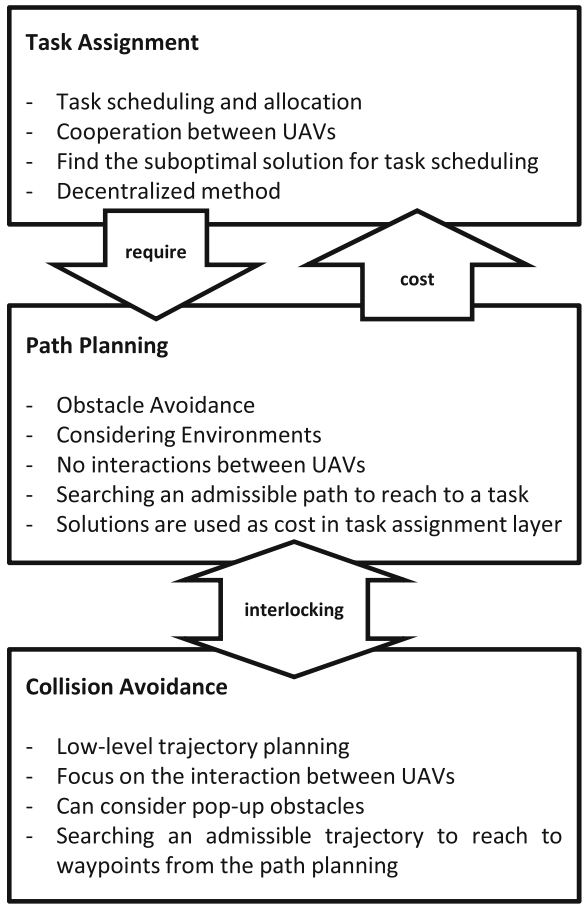

Fig. 4 Integration of task assignment, path planning, and collision avoidance layers

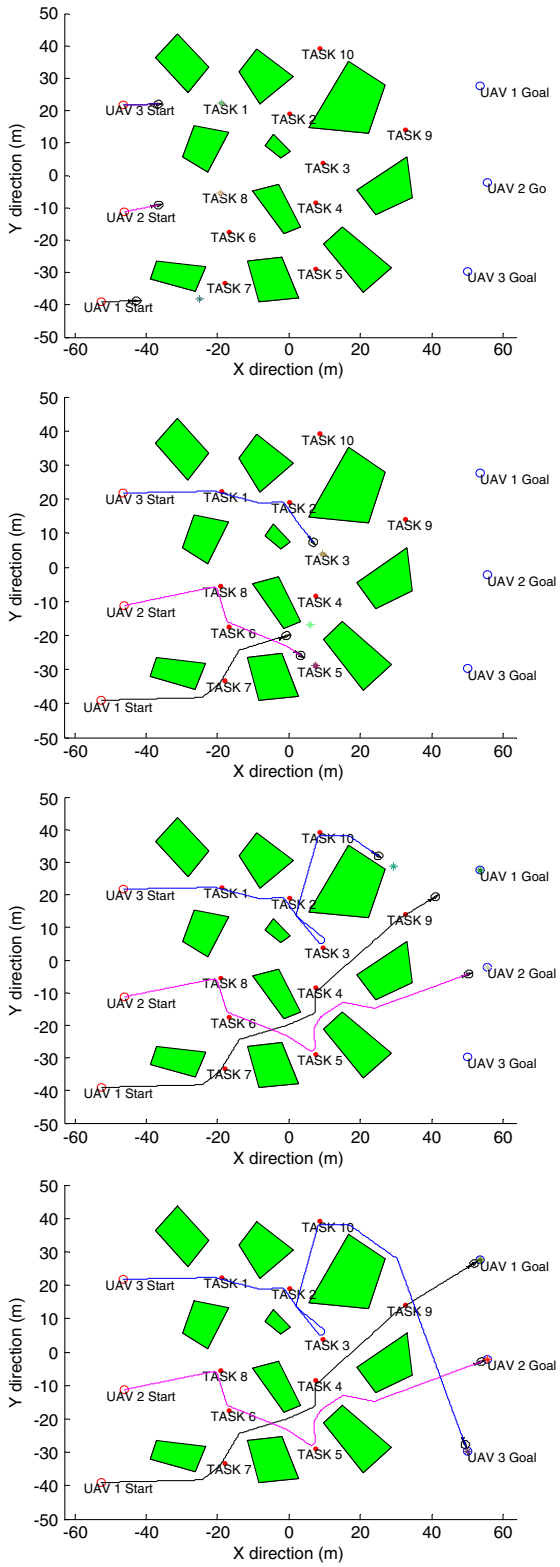

Fig. 5 Task assignment in totally known environment (t = 10,60,120, and 185 s from top)

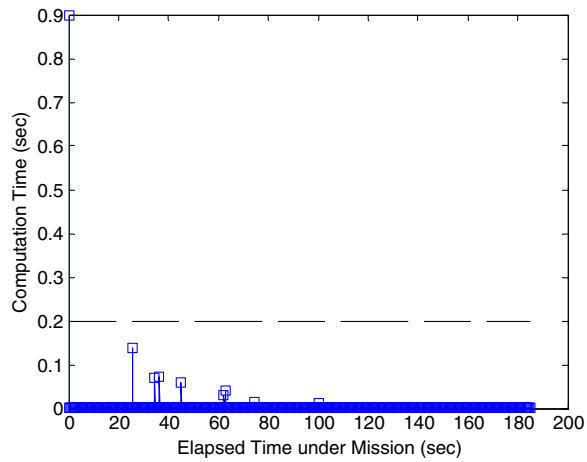

Fig. 6 Elapsed time versus computation time

7 Simulation and Analysis

To evaluate the performance of the proposed algorithm, a set of simulation is conducted. We consider a scenario with three UAVs carrying out a mission in an area with ten randomly shaped obstacles. Ten task points are arbitrarily placed in the field. As mentioned, tasks are defined as a visit to a waypoint. The margin for obstacle avoidance is two meters. Simple kinematic equations for UAVs were used in this simulation.

Figure 5 shows the simulation result. As the mission progresses, ten negotiations occurs. UAV

Fig. 7 Multi-agent testbed based on fixed-wing UAVs

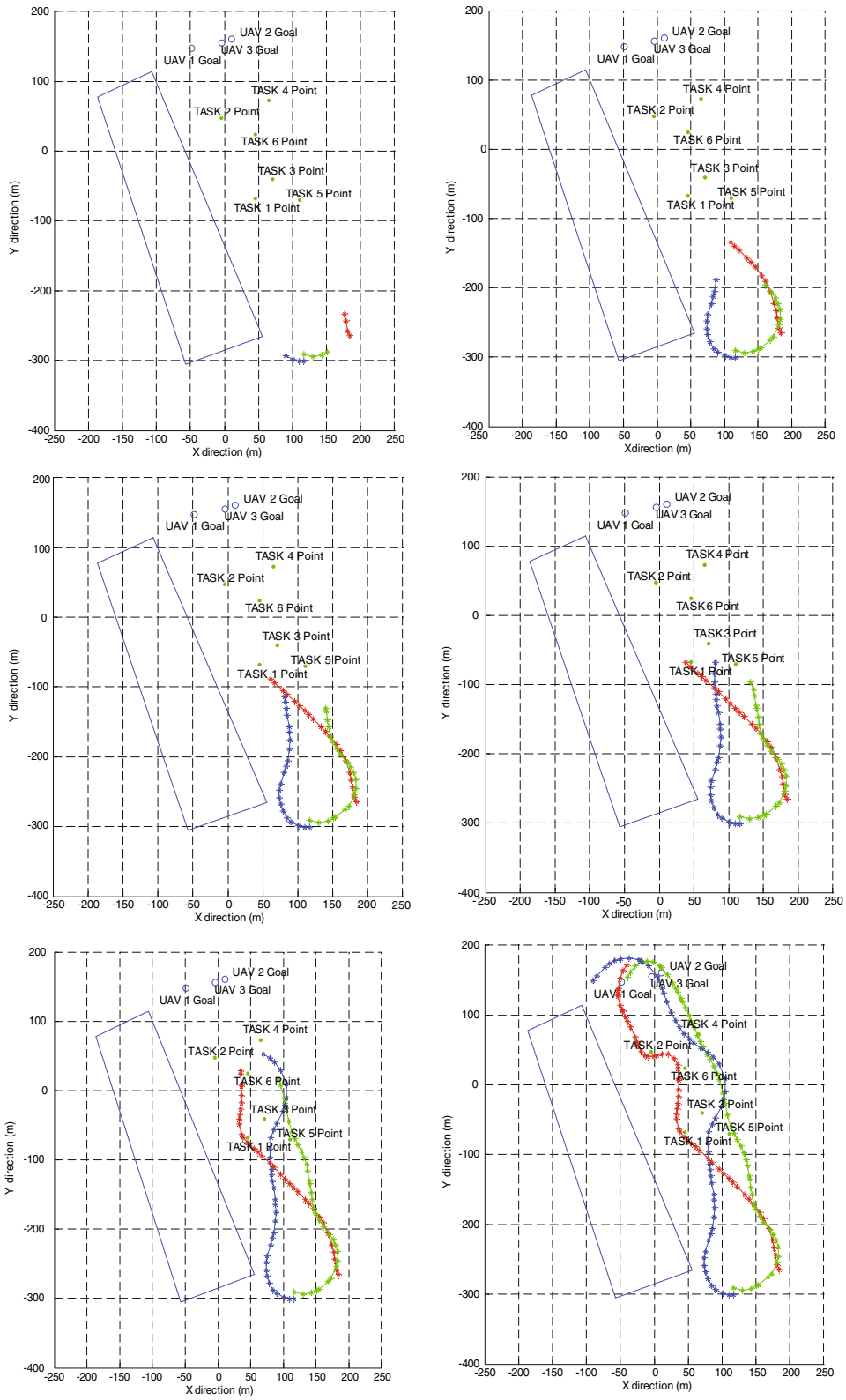

Fig. 8 A flight test result of a scenario with three UAVs and six tasks in the presence of a no-fly zone

1 is assigned with three tasks (task 4, task 7, and task 9) and participated in the whole negotiation process.

UAV #2 is also assigned with three tasks (task 5, task 6, and task 8) and UAV #3 is assigned with four tasks (task 1, task 2, task3, and task 10) during the mission. At fifth negotiation, UAV #1 sends the information of the determined task to UAV #2 and chooses to undertake task 4 because it is better that UAV #2 carries out task 5. Such a task swapping also occurs at the ninth negotiation process between UAV #1 and UAV #2.

There are nine peaks in the plot of elapsed time for negotiation process (Fig. 6). Although the maximum computation time (=0.9) of the first process is greater than the iterated time to run the proposed algorithm, it does not affect the real-time process because it is executed prior to commencing the mission. The computation time for negotiation process was quite less than the iteration time, so it shows that the proposed algorithm can be used for the real-time process. However, it is noted that the result cannot guarantee the globally optimal solution because the proposed algorithm is a greedy one and operated in a decentralized manner.

8 Flight Experiments and Validations

As the final step of validation, the proposed framework is evaluated using three fixed-wing UAVs as shown in Fig. 7. The experiment is conducted in the following steps. Each UAV communicates with the ground station through its own onboard modem. After the flight computer is initialized, the vehicle is launched using a bungee cord. After the vehicle climbs to a prescribed altitude, it begins loitering in a circular pattern until a task is assigned. When all agents are airborne and ready to accept the task commands for the mission, the ground station starts the task algorithm, which receives all agents' position and heading and sends the task commands back to each agent over the wireless network.

When the UAVs receive the task command from the ground, the onboard flight controller steers the vehicle to visit the commanded location. A task is declared completed if the UAV passes the commanded location within a prescribed bound. When the UAV receives a new task request while flying to a task location previously assigned, the flight controller commands the vehicle to fly to the newly received task waypoint, simply ignoring the previous request.

Here, we first consider a scenario of three UAVs to visit six check points for reconnaissance with a no-fly zone on the left (Fig. 8). The experiment result is shown in Figs. 8 and 9. In these plots, the trajectories of three UAVs performing the scenario as well as the task points and simulated obstacles are shown. Initially, the three agents are located at the lower right area of the map and then start flying to the upper left direction upon receiving the task commands.

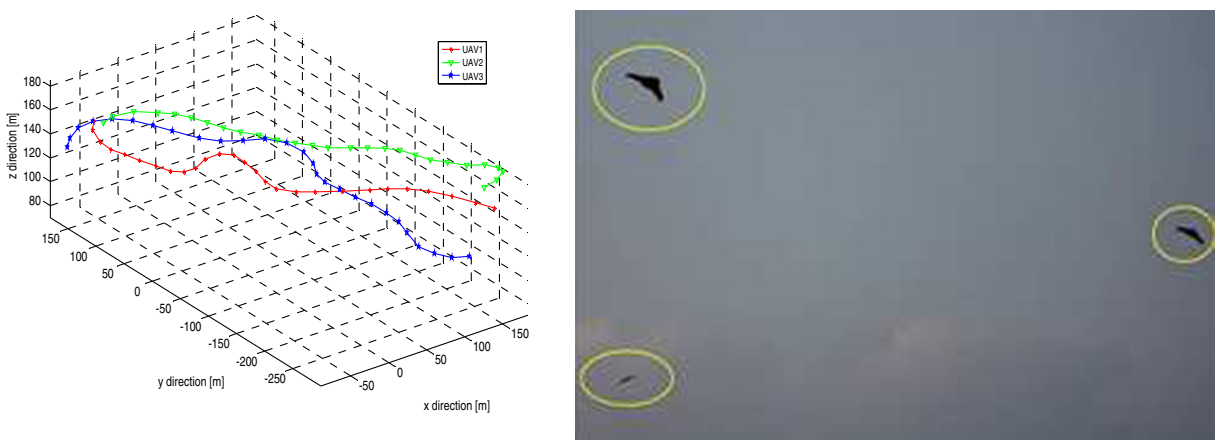

Fig. 9 A snapshot and a three-dimensional graph of the trajectories of three UAVs in a dynamic task assignment experiment

Table 2 Task assignment procedure for experiment with three UAVs

	UAV #1	UAV #2	UAV #3
Step 1	TASK 1	TASK 5	TASK 3
Step 2	TASK 3	TASK 6	TASK 5
Step 3	TASK 6	TASK 4	TASK 3
Step 4	TASK 6	TASK 4	TASK 2
Step 5	TASK 2	Finish	TASK 4
Step 6	TASK 2	Finish	Finish

Table 2 shows how the tasks are assigned to three UAVs during the mission. Based on the initial positions of the agents, tasks are initially assigned to the agents as shown in step 1. When task 1 is completed by UAV #1, the remaining tasks are assigned to all UAVs as shown in step 2. This time, UAV #3 completes task 5. The remaining tasks are again assigned to all other UAVs and this process repeats until all tasks are completed at step 7. In this example, only UAV #1 and #3 perform the tasks while UAV #2 did not have a chance to perform any of the tasks it was bidding for due to the cost higher than that of other UAVs' bids.

In the second scenario, we evaluated the obstacle avoidance capability against pop-up threat, which was not known during the initial path planning stage. Here, two UAVs fly into an area with four pop-up threat zones. The experiment result is shown in Fig. 10, where the two UAVs were commanded to perform the given tasks while avoiding

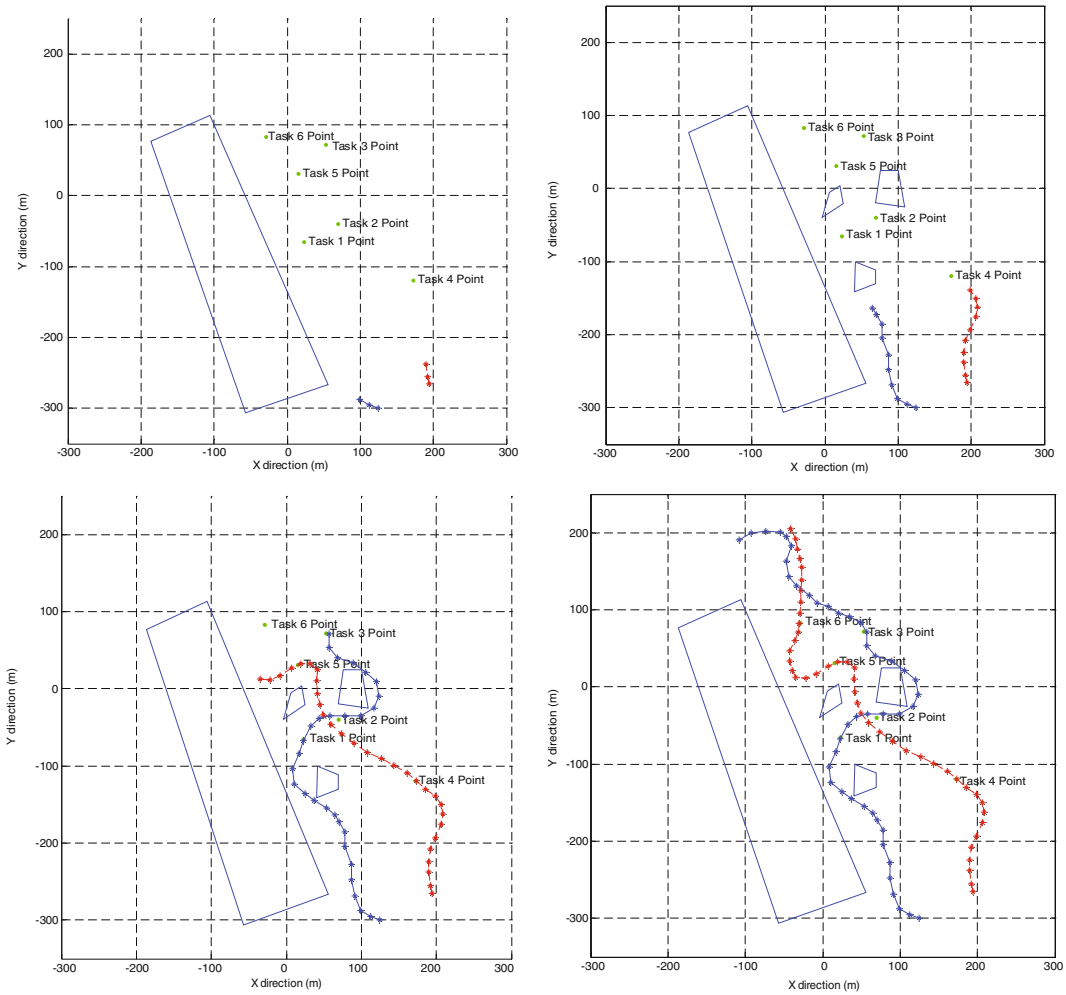

Fig. 10 A flight test result of a scenario with two UAVs with a no-fly zone and pop-up threats

the pop-up threats using the potential field-based collision avoidance algorithm.

From this experiment, the proposed framework is shown to be capable of performing a dynamic task assignment in the presence of pop-up threats as well.

9 Conclusion

In this paper, a hierarchical framework for task assignment, path planning, and real-time collision avoidance is proposed. The task assignment layer is based on a negotiation-based algorithm and the path planner is constructed by combining the shortest-path principle with A^* search algorithm. The real-time collision avoidance algorithm for pop-up threats or unmapped obstacles is based on potential field. The proposed framework is first validated in simulations and then in a series of experiments using real fixed-wing UAVs in outdoor environment. In this experiment, the group of UAVs was able to accomplish the given missions by visiting all of the task points even in the presence of known obstacles and pop-up threats. The proposed algorithm is expected to be readily applicable to various multi-UAV scenarios in real environment unlike many algorithms that function only in sterile conditions.

Acknowledgements The authors gratefully acknowledge for financial support by Korean National Research Foundation (Project number 20110015377) under Ministry of Knowledge and Economy, South Korea. The authors also gratefully acknowledge Major Yang, Gwang Jin for his help for flight experiment.

References

1. Bellingham, J., Tillerson, M., Richards, A., How, J.P.: Multi-task assignment and path planning for cooperating UAVs. Presented at the Conference on Cooperative Control and Optimization (2001)
2. Chung, H., Oh, S., Shim., D.H, Sastry, S.S.: Toward robotic sensor webs: algorithms, systems, and experiments. In: Proceedings of IEEE, 99-9, pp. 1562–1586 (2011)
3. Shim, D.H.: A dynamic path generation method for a UAV swarm in the urban environment. Presented at the AIAA Guidance, Navigation, and Control Conference (2008)
4. Schouwenaars, T., Moor, B.D., Feron, E., How, J.: Mixed integer programming for multi-vehicle path planning. In: European Control Conference, pp. 2603–2608 (2001)
5. Richards, A., Kuwata, Y., How, J.: Experimental demonstrations of real-time MILP control. Presented at the AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, TX (2003)
6. Earl, M., D'Andrea, R.: Iterative MILP methods for vehicle control problems. IEEE Trans. Robot. **21**, 1158–1167 (2005)
7. Jain, V., Grossmann, I.E.: Algorithms for hybrid MILP/CP models for a class of optimization problems. INFORMS J. Comput. **13**, 258–276 (2001)
8. Chang, D.E., Shadden, S.C., Marsden, J.E., Olfati-Saber, R.: Collision avoidance for multiple agent systems. In: 42nd IEEE Conference on Decision and Control, Maui, Hawaii, USA (2003)
9. Pettersson, P., Doherty, P.: Probabilistic roadmap based path planning for an autonomous unmanned helicopter. J. Intell. Fuzzy Syst. **17**(4), 395–405 (2006)
10. Maza, I., Caballero, F., Capitan, J., Martines-de-Dios, J.R., Ollero, A.: Experimental results in multi-UAV coordination for disaster management and civil security applications. Presented at International Symposium on Unmanned Aerial Vehicle (2010)
11. Moon, S., Shim, D.H.: Development of an efficient path planning algorithm using UAVs in cluttered environment. Presented at Institute of Control, Robotics and Systems Conference (2010)
12. Bertsekas, D.P.: An auction algorithm for shortest paths. SIAM J. Optim. **1**, 425–447 (1991)
13. Ren, W., Beard, R.W., Atkin, E.M.: Information consensus in multivehicle control. IEEE Control Syst. Mag. **27**(2), 71–82 (2007)
14. Choi, H.L., Brunet, L., How, J.P.: Consensus-based decentralized auctions for robust task allocation. IEEE Trans. Robot. **25**(4), 912–926 (2009)
15. Viguria, A., Maza, I., Ollero, A.: Distributed service-based cooperation in aerial/ground robot teams applied to fire detection and extinguishing missions. Adv. Robot. **24**, 1–23 (2010)