Planning and control for Unmanned Air Vehicle formation flight

Gautier Hattenberger, Rachid Alami, Simon Lacroix

HAL Id: hal-01979766

https://hal.laas.fr/hal-01979766

Submitted on 13 Jan 2019

Planning and control
for Unmanned Air Vehicle formation flight

Gautier Hattenberger, Rachid Alami, and Simon Lacroix

LAAS-CNRS

7, Avenue du Colonel Roche,

31077 Toulouse Cedex 4, France

Email: {firstname.name}@laas.fr

Abstract— We present a set of algorithmic solutions for a fleet of unmanned aerial vehicles evolving along in tight coordination in a constrained environment. The mission to achieve is defined by an ordered sequence of waypoints and a set of constraints to satisfy. The algorithms allow the determination of the geometric configuration of the UAVs, of the trajectories to switch from one configuration to an other, and the flight control of each individual UAV. All the algorithms are embedded within an architecture that ensures a safe and robust behavior of the formation.

I. INTRODUCTION

Deploying multiple unmanned aerial vehicles (UAVs) is very relevant for a wide range of applications, *e.g.* environmental exploration, monitoring or surveillance, intervention on specified targets... The control of such systems offers many challenging problems from both a theoretical and a practical point of view. A particular problem is the management a group of UAVs flying in tight cooperation within a reduced volume, often referred to as "formation flight". Evolving in such conditions is suited for missions that require the synchronized arrival of various UAVs on a target located in a hostile environments – typically in defense applications. UAVs formations induce specific difficulties: in particular, it calls for a *tight integration* of flight control with the decisional abilities, for instance the ones that specify the geometric organization of the UAVs given threats located on the ground.

Related work: The problem of formation flight has been studied in many ways. From the flight control point of view, classical leader-wingman configuration can be handled with proportional-integral control [1] or non-linear control [2]. A reactive behavior-based controller is presented in [3]. Trajectory optimization for large formations can be solved by centralized [4] or distributed [5] algorithms, taking into account some constraints on the shape of the formation. Contributions on the structure of the formation itself and the optimization of the data information among the aircrafts can be seen in [6]. The problem of formation reconfiguration (switching from a particular geometric pattern to an other one) is well introduced in [7]. They propose a scheme where transition trajectories are computed off-line for switching between a limited number of formation configurations. Configuration selection is performed through a finite state machine.

It is worth noting that the contributions that address planning and execution control architectures for teams of UAVs, *e.g.* [8], [9], seldom consider issues raised by formation flight.

Contributions and outline: Our work investigates a new approach, in which an "intermediate" layer receives a mission for the formation (defined by waypoints to reach and a set of tactical constraints to satisfy) from a central planning system, and sends commands to the flight control system of each UAV. This layer carries out the following functionalities:

- choice of the formation geometric pattern according to the known constraints,
- specification of the reconfiguration maneuvers,
- flight control parametrisation of each UAV, in order to achieve safe trajectories.

The choice to embed these functions in an single layer is motivated by the fact that the upper level deliberative layer is released from internal formation problems and can "manipulate" the formation as a whole. Therefore, the complexity of the planning activity is reduced, computations can be distributed amongst aircrafts, matching the constraints possibly raised by limited on-board CPU capabilities. Also, some reactions to unpredicted events (such as the detection of an initially unknown threat) can be directly processed at this level without calling for higher level planning activities – provided the event can be handled without plan modifications.

The next section precisely states the tackled problem, and presents the overall approach to solve it. Section III depicts the various algorithms involved in the management of the formation (geometric pattern selection, reconfiguration trajectories planning, control and optimization). Finally section V describes how the processes are integrated within a global decisional architecture, and distributed among the UAVs.

II. APPROACH

A. Problem statement

We assume that a mission preparation phase has been carried out off-line, on the basis of the knowledge of the terrain, the position of threats on the ground[1], and the specification of the target to reach. The output of this phase is a *mission plan*, described by an ordered sequence of waypoints to reach and a set of constraints to satisfy during the flight. Note that a mission plan may also involve several groups of UAVs (multi-formation), and specify formations split and joint maneuvers.

---

[1]Typically ground to air missiles and early warning radars.

A waypoint is defined by a unique Id, a position (latitude, longitude, altitude), a best time of arrival with a min/max interval, a nominal speed with min/max interval, a maximum load factor during turn, and possibly an associated heading. There are three type of waypoints : *flyby*, *flyover*, *flyover_with_heading* (see section III-A).

The constraints are associated to terrain areas or may hold between two given waypoints. There are two types of constraints:

- "Internal constraints" are relative to the inner organization of the formation and motions of the aircrafts. At least, a minimum and maximum distance between the UAVs should be specified to ensure a secured flight and a good communication level. It is also possible to specify that a given position in the formation should be dedicated only to UAVs equipped with a given device (*e.g.* a radar jamming module). This latter internal constraint is usually related to an external one.
- "External constraints" are caused by the context. The most common external event is the presence of an enemy air defense, such as missile site or early warning radar[2]. An other important external constraint is the width of the corridor in which the UAVs must fly.
  Split and join maneuvers are also considered as external constraints. When a split is needed, the mission plan specifies a distribution of the payload between the new formations: our system considers this as an external constraint and allocates the UAVs accordingly.

B. Principle of the approach

The problem at hand is to autonomously adapt all the UAVs actions in order to fulfill the mission plan, *i.e.* leading the UAV team to the specified waypoints while satisfying a set of constraints. This implies the specification of the formation (geometric distribution of the UAVs), the definition of the UAVs trajectories to adapt the formation to the constraints, and flight control of the individual UAVs.

We define the following terms:

Definition 1 (Formation): A formation is a set of UAVs that fly in tight coordination within a reduced volume.

Definition 2 (Configuration): A configuration is a geometric description of a formation, composed of patterns whose relative positions can be a function of the time. A configuration is typically adopted to satisfy one or several constraints.

Definition 3 (Pattern): A pattern is an invariant geometric form relating several UAVs.

Definition 4 (Slot): A slot is a position in a pattern. It can be tagged with some constraints, that allow to cast the UAV to it.

The first step of our approach is the generation of a trajectory to be followed by the group of UAVs, on the basis of the waypoints list (section III-A). Then, the formation must be specified: we start by selecting a configuration (section III-C) and the patterns that compose it. This gives a set of slots,

that must be tagged according to the constraints and assigned to the aircrafts (section III-D). Once this step is done, the formation is specified. The last step is to plan reconfiguration trajectories to safely switch from the previous configuration to the new one (sections III-E).

At the execution level, the UAVs must be controlled so as to satisfy the following constraints:

- The center of the formation is on the trajectory,
- the geometry of the formation is maintained,
- and a minimum security distance between aircrafts is maintained.

The first point is ensured by a tracking algorithm that uses a correction on the position corresponding to the relative position of the aircraft from the center of the formation. For the second and third points, corrections on speed, heading and altitude are sent to the individual UAV flight control modules. Those corrections are based on potential functions and the knowledge of the position of the other UAVs (section III-B).

III. ALGORITHMS

A. Trajectory generation

To generate the trajectory that follows the waypoint sequence provided by the mission plan, we use a simple and fast algorithm that yields a trajectory composed of straight lines and circle's arcs. Three types of waypoints are considered (Fig. 1): on a *flyover* waypoint, the trajectory crosses the waypoint with an arbitrary heading, while on *flyover_with_heading* waypoint it is imposed. On *flyby* waypoints, the trajectory doesn't need to cross exactly the waypoint and can quickly pass from a leg to an other[3]. The trajectory is built with Dubin's curves [10]. The turning radius r is given by $r = \frac{V_{max}^2}{g\sqrt{n_{max}^2-1}}$, where g is the acceleration of gravity, V_{max} is the maximum speed and n_{max} the maximum load factor. This conservative way of choosing the turning radii ensures that the aircrafts can handle the turns at maximum speed without exceeding load factor constraints.
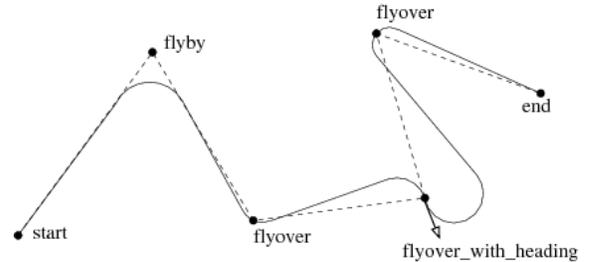

Fig. 1. The three different waypoints considered.

B. Distributed formation control

This section presents the reactive control that produces the UAVs autopilot's corrections to keep the given geometry of a formation and the minimum distance between aircrafts. This

---

[2]In which case, one of the UAV with the good jamming equipment should be placed in the right position for mutual support for instance.

[3]If the found trajectory exceeds the corridor bounds on such a waypoint, it is transformed as a *flyover* waypoint.

control is inspired by [3]. It is based on two virtual forces, whose weighted sum is decomposed in the aircraft local frame. Each UAV continuously broadcasts its position using a short range secured data link, but since the UAVs are operating in uncertain environments, one can not take for granted that each UAV know the other's positions. The algorithm must therefore tolerate occasional communication losses. We introduce the following terms (see Fig. 2.a):

- x_i, the state of UAV i.
- ΔP_i, the position of UAV i from the center of the formation (the "virtual center", located on the planned trajectory). ΔP_i is the correction that is sent to the UAV autopilot (see section V-A).
- ΔP_{ij}, the theoretical relative position between UAV i and j, equal to $\Delta P_j - \Delta P_i$.
- $\Omega(i)$, the set of UAVs whose state is known by UAV i.
- k, the number of element in $\Omega(i)$.
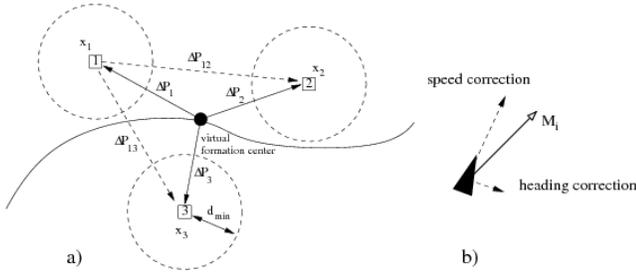- d_{min}, the security distance between two UAVs.

Fig. 2. a) The three constraints to satisfy for the reactive control: the virtual center on the trajectory, the correct geometry of the formation and the minimum distance between UAVs. b) The global reactive force decomposed in the UAV reference frame

The first force F_i imposed on the UAV is a "formation" force that keeps the global geometry of the group:

$$F_i = \frac{1}{k} \sum_{j \in \Omega(i)} \left((x_j - x_i) - \Delta P_{ij} \right)$$

If each aircraft receives at least the state of another one and if there are no independent communication loops among the formation, then the formation geometry is maintained. The more the UAVs states are shared in the formation, the more rigid the formation.

The second force provides safety reactions when two aircrafts get closer:

$$R_i = \sum_{j \in \Omega(i)} \frac{1}{(\|x_i - x_j\| - d_{min})} \frac{x_i - x_j}{\|x_i - x_j\|}$$

This anti-collision force only works if the communication between the two considered UAVs is available – one must assume that this is satisfied, which is reasonable when the UAVs get closer each to another.

Finally, the global reactive force is a weighted sum of the two previous: $M_i = a \cdot F_i + b \cdot R_i$ with $a > 0, b > 0$. Decomposed in the aircraft reference frame (Fig. 2.b), it gives

the value of the speed, heading and altitude corrections to be applied by the UAVs autopilot.

C. Configuration selection

Given a set of external constraints, the first step to ensure their satisfaction is to select an adapted configuration. This issue could be tackled as a geometric constraint satisfaction problem, of by finding in a library of pre-defined configurations the one that satisfies the constraint. The first approach could lead to heavy computations, while the second would require the considerations of too many cases to allow the determination of good solutions.
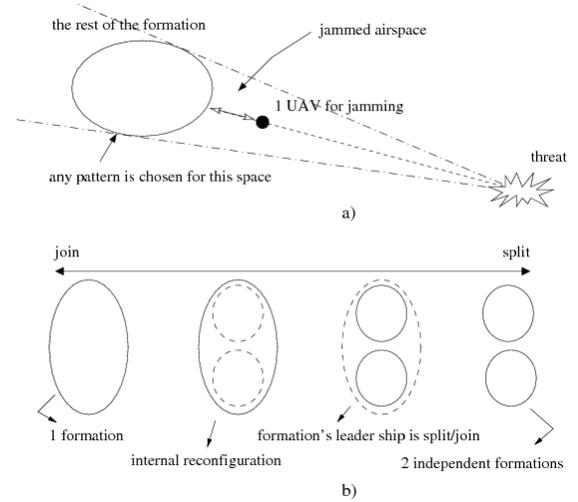

Fig. 3. a) Configuration for mutual support where the rest of the formation stay inside the jammed area. b) Configuration for a split or join maneuver with internal reconfiguration for a safe transition

To simplify the problem, we describe the formation with geometric forms and parameters. For example, in Fig. 3.a, an UAV is placed between the threat and the rest of the team for mutual support. The aircraft selected for jamming will have a higher priority during reconfiguration planning (see III-E) and the other UAVs should remain in the restricted airspace behind him, which gives a geometric relation function of the position on the trajectory. The problem is now to select a compact pattern for the hidden aircrafts. A first approach is to use generic patterns stored in a data base. It consist of classical formation patterns, such as "V" or diamond, that can be applied with homothetic transformations to match the internal constraints. A similar process can easily be defined for split and join maneuvers (Fig. 3.b).

D. Slot assignment

Once a configuration has been chosen, the slots should be assigned according to the constraints. The constraints on a slot can be the type of aircraft, a device that the aircraft must carry or a list of allowed UAVs. To solve this problem we use a Branch&Bound algorithm that can either give the best first solution or the optimal one. In either way, if there is a solution, it will be found. The benchmark for optimization is

the evaluation of the total distance to be made by all the UAVs. So we use the sum of the Euclidean distances between each aircraft and its assigned slot, in a local formation reference frame.

The algorithm is initialized by filling a table with the UAVs on the rows and the slots on the columns, each element d_{ij} in the table being the distance between UAV i and slot j. If there is a constraint on a slot (for example, only aircrafts with a jamming device can use the slot), infinite values are placed to forbid some aircraft to use the slot. The search graph is initialized with this table and a zero value for cost and bound.

At each step, we pick the best choice (the lower value) in the table of the graph's cell (UAV i and slot j). We can compute two new cells. The first is the choice "UAV i goes to slot j". The cost is increased by the value d_{ij} and row i and column j are filled with infinite values. The second is the choice "UAV i doesn't go to slot j". The cost is unchanged and element d_{ij} gets an infinite value. For each cell, a bound is computed and corresponds to a reducing estimate of the final cost.

$$
\begin{aligned}
m(i) &= \min_{j \in [1,n]} d_{ij} \\
bound &= cost + \sum_{i=1}^{n} \left\{ \begin{array}{ll} m(i) & \text{if } m(i) < \infty \\ 0 & \text{if } m(i) = \infty \end{array} \right.
\end{aligned}
$$

where n is the number of UAVs and slots. If a table is filled with infinite values without giving a solution, the corresponding branch of the graph has no solution and is dropped.

Finally, we search in the graph amongst the cells at the end of the branches the one with lower bound. If it is a solution, it is the optimal. Else, it becomes the new current cell and goes to next step. If we replace the bound of the choice "UAV i goes to slot j" by the same without the $cost$ value, the algorithm returns the best first solution. In Table I we compare the number of iterations and the final cost between the optimal and best first solution on 1000 random samples for 6 UAVs. Best first solution require more than 7 times fewer iterations than the optimal solution to provide a result, for a cost of only 6.9 % above the optimal solution cost. Our conclusion is that if on-board computation capabilities are limited, the best first solution should be use. In general, the main benefit of this solution is the almost constant number of iterations (low standard deviation of 4.1).

E. Reconfiguration trajectories

When important changes occur, such as a new configuration or a new slot assignment, it is very important that the UAVs reach their new positions safely, while keeping security distances between them. Many solutions has been studied for multi-robots trajectory planning based on linear programming [11], dynamic networks [12], probabilistic roadmaps [13] or Rapidly-exploring Random Trees [14]. These algorithms are based on structured graphs that require off-line computation or a complex optimization process. They are usually well suited for coordination in static, very constrained environments. The

	mean	std
number of iterations (optimal)	55.1	194.9
number of iterations (best first)	7.7	4.1
cost (optimal)	187.8	48.4
cost (best first)	201.6	50.1
cost increase	6.9 %	

solution proposed by [7] is not suited either because our configurations are not fully predefined so we can't make an off-line computation of the reconfiguration trajectories. We propose a fast practical algorithm based on graph search.

We assume that in a local reference frame linked to the formation, the aircrafts are free-flying objects and can move in every directions. So we are planning their relative movements inside the formation. We can work either in two or three dimensions. The algorithm is initialized with the current position of the UAVs and their final position in the formation frame, and the minimum and maximum distance between them. A first trajectory is computed for each aircraft without any constraints.

At each time step, we check the current position to detect collisions between UAVs (either too far or too close). If there is a collision at time t_i, we compute a new trajectory for one of the aircrafts. The first attempt is for the UAV with the shortest path except if it has a higher priority on his goal. If the first attempt fails, a second attempt is made with the over aircraft. In each case, between initial time t_0 and t_i, we use a spatio-temporal A* and the constraints on distances are taken into account. This means that the graph is exploring the space in all directions and the time in only one direction. The benefit is that between t_0 and t_i, the robot can stop, move backward or forward on his trajectory and leave its final position to let another robot pass through it, and then go back in place. After t_i, it is a spatial A* with no constraint that leads to the goal. The step at time t_i is re-checked until no collision remains and then it goes to next time step until all aircrafts reach their goals.

During this planning process, the airspace can be limited if an external constraint has been set on the width of the corridor, so that the UAVs do not go to far from the formation virtual center.

IV. SIMULATION RESULTS

The trajectories are planned as relative moves in the formation reference frame, but there is no guarantee on the time to reach the final formation. An estimation of this time based on the trajectory and the aircraft dynamics is reported to the deliberative layer before the execution. If needed, a new plan can be made, otherwise the reconfiguration trajectories are ready to be executed. To achieve them properly with respect to the aircrafts velocities and heading rate constraints, we need to

synchronize all elementary moves. The UAV in charge of the planning sends the new intermediate position of each aircraft and wait for synchronization signals from all of them before sending the next step.
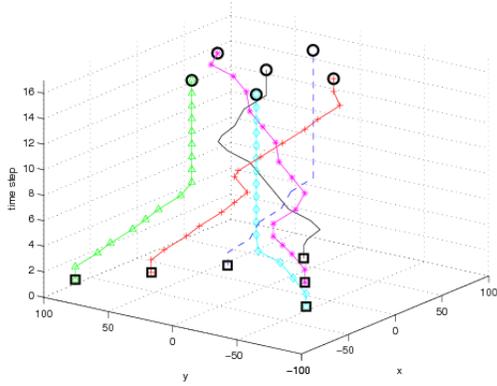

Fig. 4. Reconfiguration trajectories in formation reference frame from a "v" pattern to a more compact pattern for 6 UAVs

Fig. 4 shows a two dimensional reconfiguration for six UAVs, in function of the time (vertical axis). The initial pattern is a classical "V" formation and the final is a more compact formation.
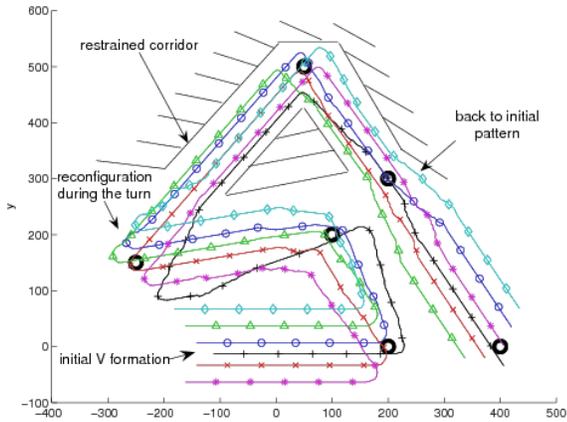

Fig. 5. Final trajectories for 6 UAVs with reconfigurations on turns to balance consumption and two changes of formation to pass through restrained airspace

On Fig. 5, we can see final trajectories for a 6-UAVs formation with reconfiguration maneuvers on each turn and two types of pattern.

V. ARCHITECTURE

The issue of organizing the algorithms within the formation is an important one. For the sake of efficiency and robustness, the organization should support both communication interruptions or UAVs failure, and satisfy stringent real-time constraints.
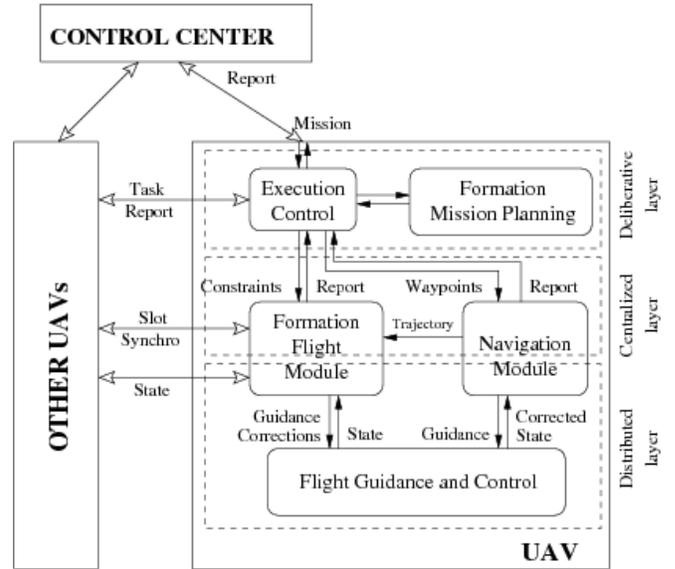

Fig. 6. Global architecture

The overall architecture designed to support our algorithms is shown Fig. 6. At its top level, the deliberative layer receives the missions to achieve from the control center, and produces a mission plan for the formation, composed of a set of waypoints and constraints to satisfy. The way this plan is produced and its execution is controlled being out of the scope of this paper, this layer is no longer depicted.

At an intermediate layer, two modules embed the set of algorithms presented in section III, and interact with the UAVs flight control module. These two modules are detailed on Fig. 7.
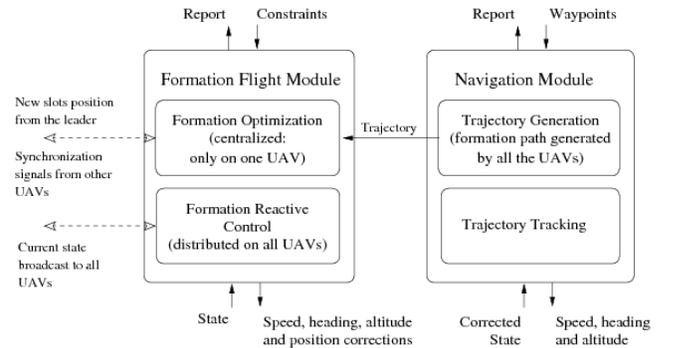

Fig. 7. Formation flight and navigation modules with internal (plain lines) and external (dashed lines) communications

A. The navigation module

The *navigation* module receives a list of waypoints from the deliberative layer. Its role is on one hand to generate the formation trajectory, which is sent to the formation flight module, and on the other hand to ensure that the formation virtual center tracks the trajectory. During the flight, a tracking algorithm inspired by the work of [15] produces speed,

heading and altitude commands that are used by the *flight guidance* module. This approach with a trajectory generation and a feedback controller for stabilization around the trajectory is depicted in [16]. In formation flight mode, each aircraft follows parallel virtual paths thanks to corrections on the state, so that the center of the formation is on the common path. There is no communication between UAVs at this level.

B. The formation flight module

The *formation flight* module is a two layers system (Fig. 7).

The upper layer is in charge of the formation's creation and of its reconfiguration, according to the constraints received from the deliberative layer. It only runs on one aircraft at a time. This aircraft sends to all the other UAVs their theoretical relative position inside the formation from the virtual center. During reconfiguration maneuvers, other aircrafts should send back synchronization signals at each step so that the reconfiguration trajectories are performed safely. There is no need to change the leader until it splits and joins an other team, or if it is lost. In either way, all planned actions should be re-planned.

The lower layer of the formation module achieves the distributed control of the formation, making use of the positions of the other UAVs (all aircrafts broadcast their state at each time step inside the formation – see section III-B).

C. The flight guidance and control module

This module is the autopilot that receives speed, heading and altitude commands from the *navigation* module and corrections (on speed, heading, altitude and position) from the *formation flight* module. It includes the inner control loop that sends commands to the aircrafts actuators. The state, attitude and position, is read from the sensors, and sent back to the upper modules with corrections, if needed.

VI. Summary and future work

We have introduced an intermediate layers that is in charge of formation planning and control, and have developed the associated algorithms. A combination of centralized algorithms and distributed reactive commands provides an efficient solution for both safe flight and tactical constraints satisfaction. This contributes to the filling of the gap between task planning and execution for teams of UAV and the autonomous flight in real environments, these problems being generally investigated separately.

Our future work will first consist in enhancing the presented algorithms – in particular, the configuration selection problem will further be studied. An other important issue to tackle is the formation mission planning, focusing on the ability to dynamically re-plan a mission during flight, *e.g.* when an initially unknown threat is perceived by the UAVs. Finally, we are in the process of implementing our algorithms on-board the UAV Lhassa (Fig. 8), and of developing a hybrid simulation framework that will allow to validate our algorithms.

Acknowledgments

Fig. 8.   The UAV Lhassa (3.0 m wingspan, 6.0 kg total)

References

[1] L. E. Buzogany, M. Pachter, and J. J. D'Azzo, "Automated control of aircraft in formation flight," in *AIAA Guidance, Navigation and Control Conference*, Monterey, CA, August 1993, pp. 1349–1370.

[2] C. Schumacher and S. N. Singh, "Nonlinear control of multiple uav in close-coupled formation flight," in *AIAA Guidance, Navigation and Control Conference*, Denver, CO, August 2000.

[3] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, December 1998.

[4] F.-L. Lian and R. M. Murray, "Real-time trajectory generation for the cooperative path planning of multi-vehicle systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, December 2002.

[5] R. L. Raffard, C. J. Tomlin, and S. P. Boyd, "Distributed optimization for cooperative agents: application to formation flight," in *Proceedings 2004 IEEE Conference on Decision and Control*, Nassau, Bahamas, December 2004.

[6] F. Giulietti, L. Pollini, and M. Innocenti, "Autonomous formation flight," *Control Systems Magazine*, vol. 20, no. 6, pp. 34 – 44, December 2000.

[7] S. Zelinski, T. Koo, and S. Sastry, "Hybrid system design for formations of autonomous vehicles," in *42nd IEEE Conference on Decision and Control, 2003*, vol. 1, December 2003, pp. 1 – 6.

[8] J. Sousa, T. Simsek, and V. Pravin, "Task planning and execution for uav teams," in *43rd IEEE Conference on Decision and Control*, vol. 4, December 2004, pp. 3804 – 3810.

[9] A. Ollero and et al, "Multiple eyes in the sky: Architecture and perception issues in the comets unmanned air vehicles project," *IEEE Robotics and Automation Magazine*, vol. 12, no. 2, pp. 46–57, June 2005.

[10] S. M. LaValle, *Planning Algorithms*.   Cambridge University Press, 2006.

[11] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *Proceedings of the European Control Conference*, September 2001.

[12] C. M. Clark, S. M. Rock, and J.-C. Latombe, "Dynamic networks for motion planning in multi-robot space systems," in *7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Nara, Japan, May 2003.

[13] P. Svestka and M. H. Overmars, "Coordinated path planning for multiple robots," *Robotics and Autonomous Systems*, vol. 23, pp. 125–152, 1998.

[14] I. Garcia and J. P. How, "Trajectory optimization for satellite reconfiguration maneuvers with position and attitude constraints," in *Proceedings of the 2005 American Control Conference*, 2005, pp. 889 – 894.

[15] E. P. Anderson and R. W. Beard, "An algorithmic implementation of constrained extremal control for uavs," in *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Monterey, California, August 2002.

[16] M. B. Milam, K. Mushambi, and R. M. Murray, "A new computational approach to real-time trajectory generation for constrained mechanical systems," in *Proceedings of the 39th IEEE Conference on Decision and Control*, December 2000.