Formation Flight: Evaluation of autonomous configuration control algorithms

Gautier Hattenberger*, Simon Lacroix, and Rachid Alami

LAAS-CNRS, University of Toulouse
Toulouse, France
firstname.name@laas.fr

***Abstract*— In military missions in hostile environments involving teams of UAVs flying in formation, it is important to get the maximum benefits of the auto-protection systems of each aircraft to enhance the global security and efficiency of the team. One way to achieve this is to select a proper configuration for the formation. In this paper, we present an approach to autonomously adapt the configuration of a formation and we focus on its evaluation within a realistic framework where each UAV is simulated independently and communicate through a network.**

I. INTRODUCTION

Formation flight is particularly suited for military applications, that require synchronizations on target arrivals and mutual support in hostile environments. In this paper, we focus on the management of a set of Unmanned Aerial Vehicles (UAVs) flying in tight coordination. One of the difficulties raised in such cases is the autonomous choice of a configuration for the formation that improves the safety and the efficiency of the team of UAVs.

Much work has been done on the control of the formation itself. Solutions have been presented for classical leader-wingman configuration based on proportional-integral control or non-linear control [1]. The behavior based control [2] have shown some capacities for handling basic reconfigurations and obstacles avoidance. Algorithms for trajectory optimization have been presented with a centralized [3] or a distributed [4] solution. Even if some contributions have been proposed on the configuration of the formation [5] or on the reconfiguration problem [6], as noticed in [7], they do not tackle the problem of the choice of the configuration. The control architectures dedicated to mission planning for teams of UAVs do not consider this issue either [8].

This work is focused on the autonomous adaptability of the configuration of a team of UAVs flying in formation. A configuration is defined by a set of *slots* corresponding to the relative position of the aircrafts. The mission is defined by a list of *waypoints* and a set of *tactical constraints* (distances, threats…). Our algorithms lie in an intermediate layer between the mission planning system and the autopilot of each UAV. This approach is motivated by the fact that the deliberative level is released from "internal" formation problems and so, can manipulate the team as a whole. Our layer is in charge of the following functionalities:

- the choice of a configuration according to the constraints and the environment,
- the planning of reconfiguration trajectories for safe transitions between two configurations,
- and the flight control loop to achieve the coordination inside the formation.

We have introduced a system to carry out those tasks in [9], [10]. Here, we present and analyze some realistic simulation results and algorithms performance tests.

Sections II and III sumerize our previous work. It precisely states the problem and then presents our approach to solve it, especially the global update process of the formation's configuration. Section IV presents simulation results. Some perspectives conclude the paper.

II. PROBLEM STATEMENT

A. Mission definition

Considering a group of UAVs flying in formation, we assume that a mission planning phase has been carried out off-line on the basis of prior knowledge on the environment and the tactical situation. The outputs of this phase are an ordered list of waypoints, a set of tactical constraints and a set of known threats.

B. Waypoints

A waypoint is defined by a position, 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. Waypoints are produced during the planning phase and can not be changed by the formation layer. The trajectory generated on the basis of these waypoints is a succession of straight lines and circle's arcs.

C. Threats jamming model

Threats are defined by a type, a position and a range. The two types considered are *Early Warning* radars and missiles sites (*Track & Fire*).

Early Warning (*EW*) are radars used for detection. They are permanently active and can be detected at a very long range: their position is known during the planning phase. *EW* can be jammed using a specific device, which is the only

* The authors would like to thank *Dassault Aviation* for their support to this work.

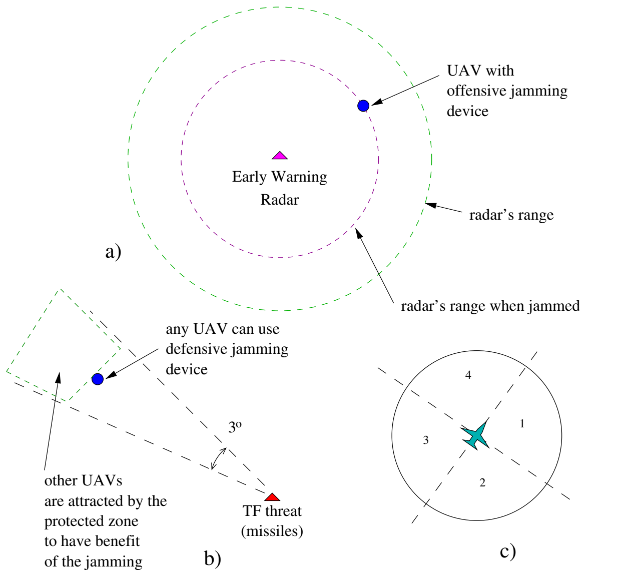

Fig. 1. a) Effects of offensive jamming on *EW* radars. b) Defensive jamming on *TF* threats. c) Four sectors defined for the UAVs.

payload that can be carried by one aircraft if we consider small military UAVs. These aircrafts will be referred as *offensive jammers* (*OJ*). The effect of an *OJ* is to reduce the range the radar (see Fig. 1.a).

Track & Fire (*TF*) systems are sol to air missiles sites controlled by *EW* radars. They usually have reactive behaviors and their position can't be well known by advance. If they use radar-guided missiles, the threat can be jammed using a light protection device called *defensive jammer* (*DJ*). The effect is to blind the guiding system of the *TF* threat in the direction of the jammer in a cone of aperture of 3° (see Fig. 1.b). All aircrafts behind the jammer inside this cone will benefit of the protection. We assume that all UAVs in a formation carry a *DJ*.

For each UAV, we define *sectors* (see Fig. 1.c). Four *EW* can be jammed inside each sector with an *OJ*, and a single *TF* threat can be jammed in a given sector with a *DJ*.

D. Constraints

Tactical constraints hold between two given waypoints. The constraints that are considered in section IV are a minimum distance between the UAVs to ensure a secured flight and the width of the corridor in which the UAVs must fly. Other constraints, such as threat priority or split/join maneuvers, can be found in our previous work [10].

III. THE APPROACH FOR CONFIGURATION CONTROL

A. Global Approach

The global approach for configuration control is first to initiate the system by computing a *timeline* (cf. III-A.1) and an *OJ* allocation (cf. III-A.2). Then, at each control step the configuration of the formation is updated based on the allocation, the current interval in *timeline*, the detected *TF* threats and the position of the formation on the trajectory. As shown Fig. 2, the system is updated when constraints, threats or waypoints are added or removed. Some of these updates can lead to a new initialization of the *OJ* allocation or the *timeline*.

Once an adapted configuration has been chosen, a distributed algorithm based on potential field [2], [10] ensures the control of the formation. The robustness of such an approach is evaluated in section IV-D.

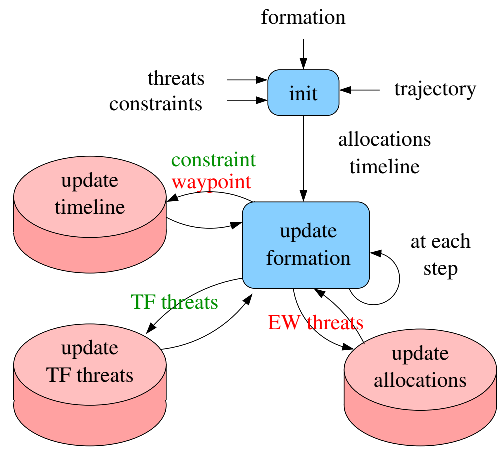

Fig. 2. The global update process. Red elements require a new initialization, while green elements do not.

1) The timeline: The constraints are defined between two waypoints. The *timeline* is a structured representation of the set of constraints other the whole trajectory. It is composed of intervals that contain the most restrictive information for each type of constraints. The parameters for configuration selection and reconfiguration trajectories computation are taken from the current interval in the *timeline*.

2) The Offensive Jammers Allocations: For *OJ* allocation, we have to consider the positions of the *EW* threats, their range, the trajectory and the UAVs sectors constraints. We define the allocation A for one jammer as a set of threats whose angular position are connex, that lie in the range of the formation and that respects the limitation on sectors and maximum aperture (see Fig. 3). The aperture of A is the biggest angle made by the center of the formation and two threats in A. The protection is maximal when the aperture is zero (one threat in A) and minimal for an aperture of 180°. So, we can define a quality criterion Q for an allocation A as the ratio between the aperture and the number of threats in A. The global *OJ* allocation finds the optimal set of allocations A that minimize the sum of Q with the minimum number of jammers [9]. The algorithm is based on *Branch & Bound* [11], *i.e.* a graph search where each node produces two branchs (a choice and its opposite) and where each iteration starts from the most promising node to the solution.

3) Motion Modes: *EW* and *TF* threats have different behaviors and the system knowledge on these threats is not the same either. So, we propose to have a different control of the configuration according to the kind of threats to be treated. We define three modes:

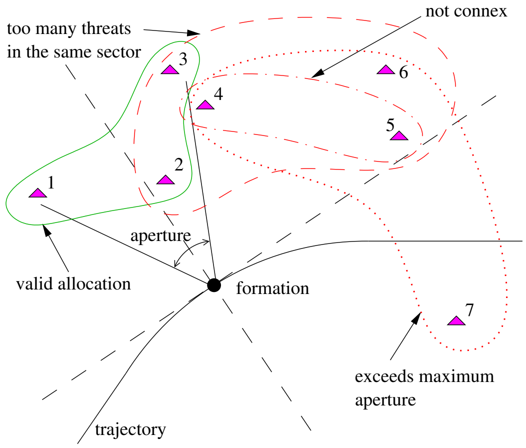

Fig. 3. Example of valid (green) and not valid (red) allocations for a maximum aperture of 90 °.

- *reactive mode* is a behavior based motion mode using a potential field. It is associated with the *TF* threats. The forces are designed to grant mutual support for the UAVs when they are using their *DJ*. It is also the default motion mode.
- *parametric mode* is used to place the jammers according to the *OJ* allocation in order to create the best protection for the rest of the formation. This mode is combined with the *reactive mode*.
- *planned mode* is used to execute reconfiguration trajectories. The trajectories are computed when important changes occur in the formation (positions or slot allocations). This mode takes control over the two others.

B. Configuration Update Process

The first step is to compute the *OJ* allocation and the *timeline* as presented Fig. 2, on the basis of the formation and the reference trajectory.

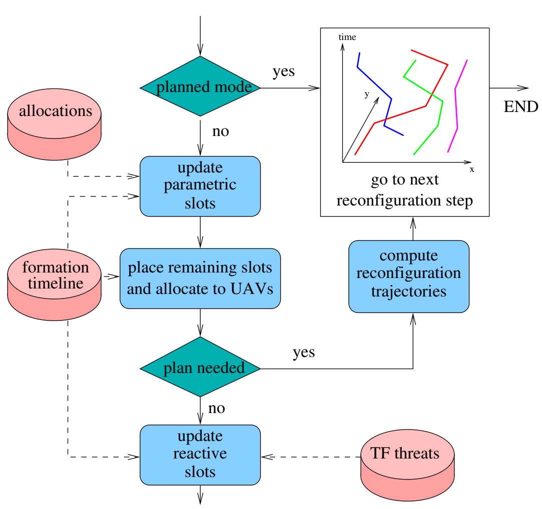

Fig. 4. Formation update process is running on only one UAV in the formation.

Then, at each step, the sequence shown Fig. 4 is achieved. If the formation is in planned mode, the reconfiguration trajectories are executed until the final configuration is reached or if the planned mode is interrupted to check if a better final configuration can be found. If not in planned move, the parametric slots are updated based on the current allocations, the *timeline* and the ressources of the formation (the number of *OJ*). Then, the remaining slots are placed and the resulting set of slots is allocated to the UAVs using a *Branch & Bound* algorithm. The next step is to check if a computation of reconfiguration trajectories is needed. This is the case if one of the UAV has to move far from its current position. So, if a new trajectory planning is needed, an algorithm based on A* [10] is run and its output is executed. Else, the slots in reactive mode move according to the potential field created by the formation and the environment [9].

IV. SIMULATION RESULTS AND PERFORMANCE ANALYSIS

A. Reconfiguration Trajectories Performance

The algorithm that computes reconfiguration trajectories is the most greedy in terms of computations. It finds trajectories in the local reference frame of the formation, and guarantees that if a solution is found, a minimum and a maximum distances between the UAVs are respected at each step. The main advantage is that it works from any initial configuration to any valid final configurations, so the formation is not limited to a finite number of configurations. The drawbacks are that neither the UAVs' dynamics, nor the global trajectory, are taken into account. This implies that the time needed for a whole reconfiguration can only be guessed from the number of steps. An other limitation due to the A* algorithm is that the computation time is not guaranteed either.

In order to avoid deadlocks in the update process, the maximum numbers of nodes explored by the A* for one trajectory is limited, and so is the number of reconfiguration steps for the whole formation. If no solution can be found, the configuration remains unchanged and a new attempt is possibly made at the next formation update with different initial and final conditions (see "fail rate" on Fig. 6).

Fig. 5 shows the mean computation time with its standard deviation (std) for a set of random initial and final configurations placed in a 200 meters radius sphere. The minimum distance between UAVs is 40 meters and the planning step 8 meters. Despite important std values, computations are fast enough for real time applications.

Fig. 6 shows the mean computation time for various planning steps in the same conditions. It clearly appears that a larger step improves the computation time, since there are less steps to reach the goal. But this result hides the fact that bigger margins should be taken for the configuration so that the UAVs won't exceed the minimum distance between two reconfiguration steps. In other words, a large planning step should be used to improve computation time only if a very tight formation flight is not needed.

This figures have been established with an implementation in C++, running on an Intel Centrino 1.6 GHz with 512 Mo of RAM. For each step and each number of UAVs in the formation, we ran between 600 and 800 random configurations.

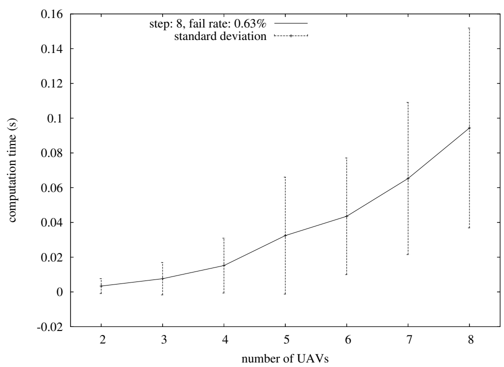

Fig. 5. Mean computation time with standard deviation for a planning step of 8m and a minimum distance between UAVs of 40m.

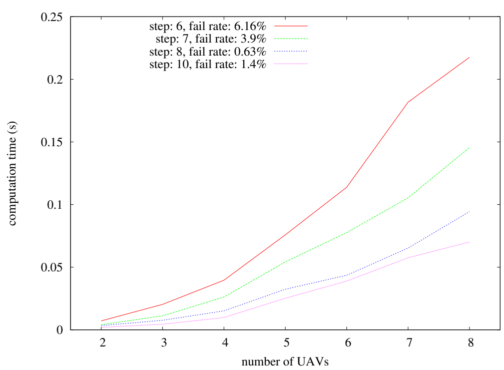

Fig. 6. Comparison of the mean computation time for different planning steps with a minimum distance between UAVs of 40m.

B. Realistic Simulation in Hostile Environment

We present a simple scenario with 4 UAVs flying in the range of one *EW* radar and two *TF* threats. The situation is shown on the Graphical User Interface used to control the formation (Fig. 8). We aim to evaluate a realistic simulation, which means with realistic dynamics, threat models and the the actual software architecture. Fig. 7 shows the simulation setting where each UAV is running on a single computer and is communicating through the network. Our system is supported by the *LAAS* architecture [12], including the functional layer and the execution controller that would run on real UAVs.

The flight control of the formation is the combination of a trajectory tracking system [13] in a navigation module that provides speed, heading and altitude setpoints and corrections on position, speed, heading and altitude that come from a formation module. On Fig. 9, the color lines ended by a sphere represent the relative position of the UAVs in the formation, so the four spheres are ideally merged. In this case, the tracking algorithm has properly placed the spheres on the trajectory but the green UAV is too far ahead, so the speed correction (the red bar above) forces it to slow down while the others speed up.

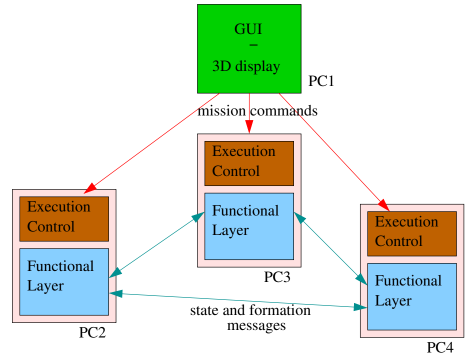

Fig. 7. The realistic simulation setting. Each UAV is simulated on a different computer, possibly hardware-in-the-loop.

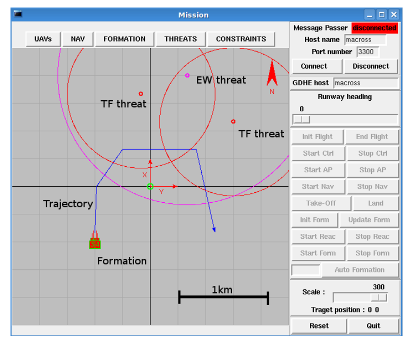

Fig. 8. User Interface for mission control. The nominal speed of the UAV is 12m/s. The scale with high speed military UAVs is respected.

In this scenario, there is only one *OJ*, the yellow UAV. On Fig. 10, we can see that the formation is performing a reconfiguration so that the yellow aircraft is placed between the *EW* radar and the rest of the formation. The range of the jammed radar is shown by the green circle's arc. Here, the system was not asked to begin the reconfiguration before entering the radar's range, so some of the UAVs have been detected during a few seconds before the *OJ* reaches his jamming position. This highlights the need to anticipate the allocation of the jammers, or at least provide a smooth transition between two allocations if the formation is in a very constrained environment. At the same time, the green

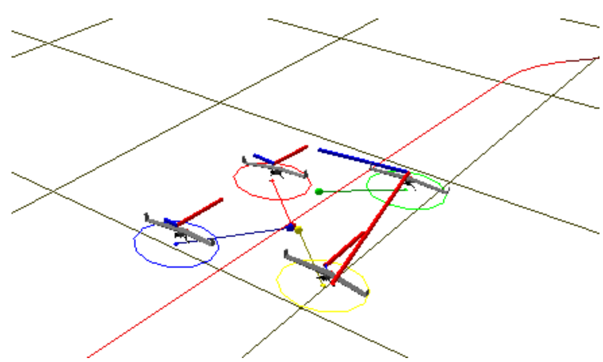

Fig. 9. Trajectory following. Red bars are speed corrections, blue bars are heading corrections. The size of the UAV is magnified ten times for a better view.

UAV is jamming the *TF* threat (shown by the green line) and thus protects all the other UAVs.

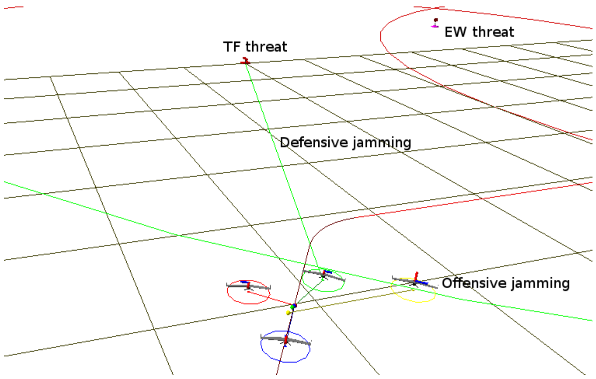

Fig. 10. The formation during reconfiguration maneuvers (jamming in green).

A comparison between Fig. 10, 11 and 12 clearly shows the evolution of the *OJ* (yellow UAV) in parametric mode to keep a position between the *EW* radar and the rest of the formation.

On Fig. 11, the other aircrafts are in reactive mode. We can see the reaction of the blue UAV to avoid the red aircraft with an important heading correction (blue bar). Even if the minimum distance between aircrafts is not exceeded, this situation shows that we need important security margins. Here, the actual minimum distance used by the system for configuration control is twice the distance specified by the operator and shown by the circles around the UAVs. A special attention should be paid to find the good trade-off between a close formation and a secure flight.

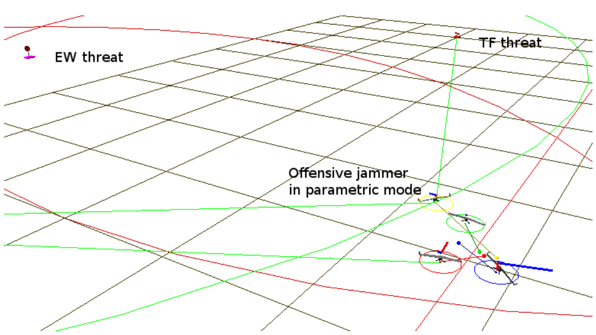

Fig. 11. Evolution of the *OJ* in parametric mode

A similar situation is presented Fig. 12 in planned mode. When the aircrafts start a new reconfiguration step, important corrections are needed to reach the correct configuration. As the vehicles dynamic is not explicitly taken into account, the overall stability and rapidity of the system only rely on the guidance system of the aircrafts.

Finding a valid criterion to evaluate the benefit of adapted configuration is still to be done. We consider some criterions based on maximum time of radar exposure and success rate on complex situations for instance.

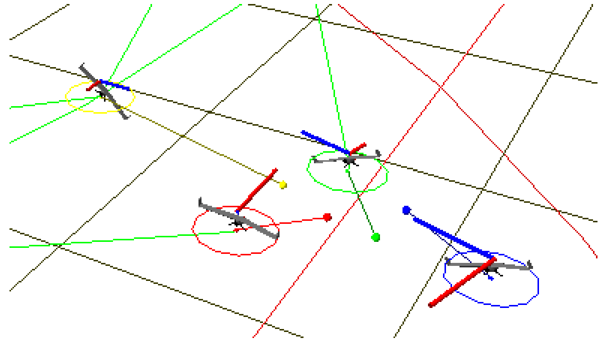

Fig. 12. The formation starts a reconfiguration

C. Formation Flight with many UAVs

We have tested the system in its limits for the number of UAVs, *i.e.* eight aircrafts. As expected the computation of the reconfiguration trajectories uses the full CPU ressources of the leader, creating some delays. The execution is delayed too, since it is more difficult to have all the UAVs stabilized on their slots as shown Fig. 13. The figure also shows an unexpected result: with many aircrafts, they naturally tend to use the third dimension while they almost allways stay in a plan with 2 or 4 UAVs. In this test we use two *OJ* (the yellow and cyan UAVs), so the formation was kept well protected almost all the time. Yet, our algorithms have shown a real loss of performances for larger teams.

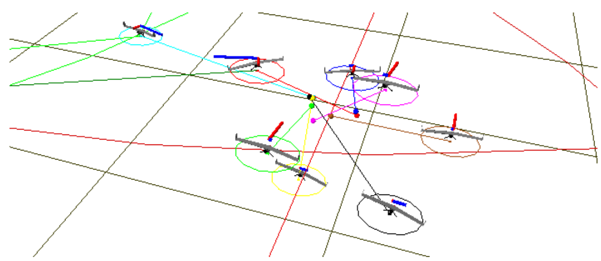

Fig. 13. Simulation with 8 UAVs, including 2 *OJ*

D. Communications

The communication between agents is one of the most critical part of formation control system [14]. Our UAVs have no other means to know each other positions than to broadcast messages inside the formation, using YARP [15] as a communication system. In a military mission, stealthiness is very important and we can assume the enemy will try to detect or even jam the communications. So, we need a system tolerant to occasional communication failures.

Our control system, presented in [10], is based on the work of [2]. This solution is known to be simple and efficient. Its robustness is evaluated as follow: a team of UAVs follows a reference trajectory and the mean deviation from the desired position is recorded. Fig 14 shows the mean deviation along the whole trajectory for different percentage of lost communication data. The formation remains stable until 60%. At 80%, collisions started to occur and the

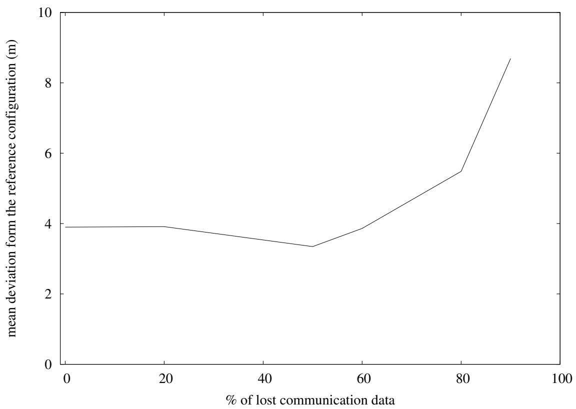

Fig. 14. Mean deviation from a reference comfiguration while following a trajectory (minimum distance between UAVs: 40m).

formation was unstable at 95%. Data are exchanged at around 5Hz. The reference configuration was a fixed square pattern.

In our application, we also need to broadcast information concerning the configuration control. As we have a centralized approach for this control, only one aircraft sends commands with the slots and the jamming allocation (*OJ* and *DJ*) of all the UAVs. It is possible to reduce the size of these messages by sending the jamming data only to the concerned UAV, but they need to know all the slots for the reactive control loop.

V. CONCLUSIONS AND FUTURE WORK

We proposed a realistic simulation framework to evaluate our system, involving a functional architecture used in real experiments and real network communications. The results have shown the benefits brought by the autonomous adaptation of the configuration in formation flight, and its limits as well. The algorithms developed to solve our problem have proved their ability for real-time applications. Most of the difficulties that appear during the tests are due to the choice of the parameters for the algorithms and security coefficients. Still, with further analysis, we could set some of them automatically.

Future work will concentrate on the validation of our system on even more realistic simulations and experiments. We are working in cooperation with *Dassault Aviation* to implement our system in the Artemis simulation framework (Fig. 15) that provides realistic scenarios and environments, and above all, a mission planning system for multiple formations. We will use this system to evaluate the split and join maneuvers and to have precise data of the benefit of our approach for military missions.

An other focus is to test the system in real experiment using small hobby aircrafts equiped with the *Paparazzi* system [16]. The test flights to validate the autonomous navigation system have recently started. We aim at a three aircrafts flight with the formation control layer running on ground station and controlling remotely the navigation system, since the payload is very limited.

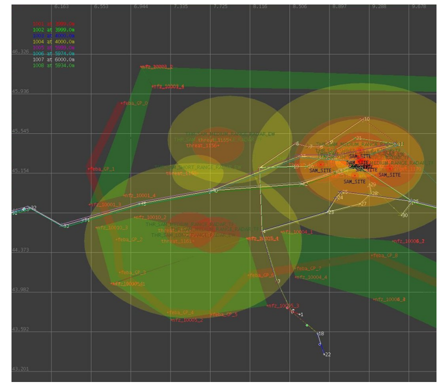

Fig. 15. Artemis simulation GUI showing 2 teams of 4 UAVs attacking a target from the south and the west.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] R. L. Raffard, C. J. Tomlin, and S. P. Boyd, "Distributed optimization for cooperative agents: application to formation flight," in *Proceedings of 43rd IEEE Conference on Decision and Control*, Nassau, Bahamas, December 2004.

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

[6] 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.

[7] Y. Q. Chen and Z. Wang, "Formation control: a review and a new consideration," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS'05)*, 2-6 Aug. 2005, pp. 3181–3186.

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

[9] G. Hattenberger, R. Alami, and S. Lacroix, "Autonomous configuration control for UAV formation flight in hostile environments," in *Proceedings of the 6th IFAC symposium on Intelligent Autonomous Vehicles (IAV'07)*, 2007.

[10] ——, "Planning and control for unmanned air vehicle formation flight," in *International Conference on Intelligent Robots and Systems (IROS'06)*, Beijing, China, 2006.

[11] A. H. Land and A. G. Doig, "An automatic method for solving discrete programming problems," *Econometrica*, vol. 28, pp. 497–520, 1960.

[12] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand, "An architecture for autonomy," *International Journal of Robotics Research*, vol. 17, no. 4, pp. 315–337, April 1998.

[13] S. Park, J. Deyst, and J. P. How, "A new nonlinear guidance logic for trajectory tracking," in *Proceedings of the AIAA Guidance, Navigation and Control Conference*, August 2004.

[14] P. Seiler and R. Sengupta, "Analysis of communication losses in vehicle control problems," in *Proceedings of the American Control Conference*, vol. 2, 25-27 June 2001, pp. 1491–1496.

[15] G. Metta, P. Fitzpatrick, and L. Natale, "YARP: yet another robot platform," *International Journal on Advanced Robotics Systems, Special Issue on Software Development and Integration in Robotics*, vol. 3, no. 1, pp. 043–048, 2006.

[16] P. Brisset, A. Drouin, M. Gorraz, P.-S. Huard, and J. Tyler, "The paparazzi solution," http://www.recherche.enac.fr/paparazzi/papers.html, ENAC University, Tech. Rep., 2006.