



The cooperation between Unmanned Aerial Vehicles using a mission planner

Tan-Viet-Anh Truong, Gautier Hattenberger, Catherine Ronflé-Nadaud

► To cite this version:

Tan-Viet-Anh Truong, Gautier Hattenberger, Catherine Ronflé-Nadaud. The cooperation between Unmanned Aerial Vehicles using a mission planner. INDIN 2013, 11th IEEE International Conference on Industrial Informatics, Jul 2013, Bochum, Germany. pp 797-803, 10.1109/INDIN.2013.6622986 . hal-00934293

HAL Id: hal-00934293

<https://enac.hal.science/hal-00934293>

Submitted on 3 Feb 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The cooperation between Unmanned Aerial Vehicles using a mission planner

Tan Viet Anh Truong and Gautier Hattenberger and Catherine Ronfle-Nadaud
Ecole National Aviation Civil (ENAC)
UAV team

7 Avenue douard Belin, 31055 Toulouse Cedex 04, France

Email: {tan-viet-anh.truong,gautier.hattenberger,catherine.ronfle-nadaud}@enac.fr

Abstract—The goal of the paper is to perform a cooperation between multiple Unmanned Aerial Vehicles via a mission planner. This mission planner is a generic one which could be used not only for UAV domain but also for other human-machine systems such as maritime, automotive and manned aerial vehicle (MAV). It is a completed mission planner including the high level task planning and low level of collision avoidance. Our first scenario shows the cooperation between UAVs during doing tasks when the number of UAVs is greater than that of tasks. The later scenarios will be the cooperation between two types of UAV: fixed-wing aircraft and rotor-craft for different types of task. This mission planner contributes a new module to both simulation platform and real demonstrator of relevant projects.

I. INTRODUCTION

In UAV domain, the mission planning issue has been studied longtime ago. They focused not only on trajectory planning [5], [9] but also on task planning [3], [8]. For one agent, we plan to do all the sequence of task at the desired time [4]. The planning is more complicated when we have many tasks to do, then multi agents have to be considered [6]. Recently, the online task planning [10] is also conducted after dealing with the offline one [7]. The goal of this paper is to present an online mission planner for multi-UAVs. This mission planner contributes a module to Paparazzi platform. Paparazzi is an open source platform used to control and simulate offline/online one/multi UAV for different types of UAVs as well as tasks [1], [11]. So with this new module, the Paparazzi users could perform different kinds of the cooperation between multiple UAVs.

The cooperation of multiple UAVs has been studied few years ago. Among different problems, the mission planning is merged as the main activity. For this problem, we resolve how to allocate available tasks to available aircrafts at one moment of time. Many algorithms have been proposed for mathematical and applied study. In this paper, we propose our mission planner developed by ENAC. This mission planner is used for planning the cooperation between multiple UAVs. The UAV can be the fixed-wing aircraft or rotor-craft depending on the goal type. All the simulation were done by using the Paparazzi simulation platform. The mission planner is integrated as a module in this platform. In the first simulation, we dealt with the scenario of three aircrafts doing two surveillance tasks.

All our scenarios aim the cooperation between UAVs, the autonomy of themselves, and the efficiency of distributing UAVs to tasks.

The paper is organized as following. We first describe our mission planner in section II. The demonstrator architecture and scenario are presented in section III and IV respectively. Finally, we show the simulation results in section V.

II. MISSION PLANNER

A. Mission formulation

We define the mission as the following data structures:

Mission = {Waypoint, Task, Aircraft}

Aircraft = {Aircraft Status, Sensor, Task}

Task = {Task Position, Task Status, Goal}

Waypoint = {x, y, altitude}

Aircraft Status = {UNSTARTED, WORKING, ALIVE, DEAD}

Task Status = {UNALLOCATED, ALLOCATED, DONE, FAILED}

B. High level task planning

Let's consider that we have a set of aircrafts and a set of tasks. The insertion of tasks to aircrafts is based on the time to go from the current position of the aircrafts to the position of tasks and the priority of these tasks. So the multiple criteria function for task allocation is described as the following:

$$J_{j(j=1..N)} = \min_{i=1..M} (\sum (t_{ij}, p_j)) \quad (1)$$

where N and M are the number of tasks and aircrafts respectively, p_j is the priority factor of the task j and the t_{ij} is the travel time to go from the position of the aircraft i to the position of the task j .

The mission planner algorithm is shown by Algorithm 3 in which we use task planning Algorithm 1 and task replanning Algorithm 2. Our current mission planner works in the case where the number of tasks N is smaller than the number of aircrafts M . The generic one will be studied in the scope of this research.

C. Low level planning of collision avoidance

The "TCAS" (Traffic Collision Avoidance System) technique [2], [11] is used for low level of mission planner including the collision avoidance between UAVs. As the first

Algorithm 1: Task planning

```
input : mission, task, aircraft data structure
output: Allocating tasks to aircrafts
// Loop for  $N$  tasks
while  $j < N$  do
  if  $task[j].status = UNALLOCATED$  then
    // Plan for  $M$  UAVs
    for  $i \leftarrow 1$  to  $M$  do
      if ( $aircraft[i].status \neq WORKING$ ) &
        ( $aircraft[i].status \neq DEAD$ ) then
        | continue;
      end
      CostInsertTaskToPlan( $aircraft[i], task[j]$ );
      MinimizeCost( $aircrafts, task[j]$ );
    end
    InsertTaskToPlan( $aircraft[i\_best], task[j]$ );
  end
end
```

Algorithm 2: Execution and task replanning

```
input : mission, task, aircraft data structure
output: Updating and Re-allocating tasks to aircrafts
// Loop for  $N$  tasks
while  $j < N$  do
  if  $task[j].status \neq DONE$  then
    // Replan for  $M$  UAVs
    for  $i \leftarrow 1$  to  $M$  do
      if ( $aircraft[i].status = DEAD$ ) &
        ( $task[j].status = FAILED$ ) then
        | RemoveTaskFromAircraft( $aircraft[i],$ 
        |  $task[j]$ );
      end
    end
  end
  Do Algorithm 1
end
```

Algorithm 3: Mission planner

```
input : XML mission description data structure file
output: Allocating tasks to aircrafts

Step 1 Parser XML mission file to mission data
structure;
Step 2 Initialization:
  Sending all aircrafts to their standby points;
Step 3 Start planner:
  Check status of tasks and aircrafts;
  Optimize the cost of each task allocation to
aircraft;
  Allocate tasks to aircrafts;
Step 4 Execution and update planner:
  Update the status of tasks and aircrafts;
  Remove failed tasks from aircrafts;
  Return to Step3;
```

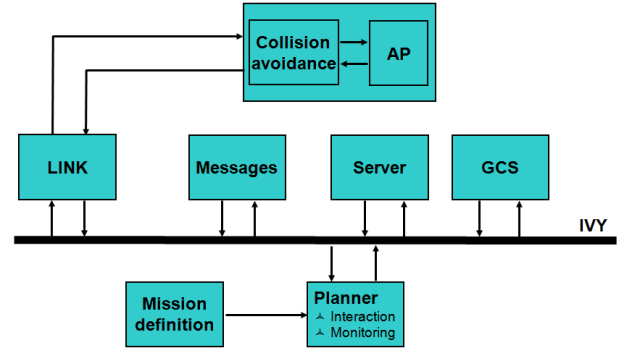


Fig. 1. The architecture of our demonstration based on Paparazzi platform.

step, we used the vertical collision avoidance module. It means that when there is a conflict between two UAVs, they will change their altitudes to guarantee their desired trajectories. Another collision avoidance method such as horizontal collision avoidance (based on the potential force field method) will be applied to this module as the next steps.

III. UAV DEMONSTRATION ARCHITECTURE

The Figure 1 shows our demonstration architecture of the cooperation between UAVs. This architecture consists of the Paparazzi modules in dark green. The paper focus on the Collision avoidance (using “TCAS”) and Planner module that we have developed for our Paparazzi platform. “TCAS” is explained in the previous section and Planner is the mission planner module integrated in paparazzi.

Each aircraft is equipped with a Collision avoidance and an autopilot (AP) on board. The connection between GCS (Ground Control Station) and onboard aircraft is based on LINK module via Messages system and the IVY bus [12]. We send and bind messages on the IVY bus between Server, GCS and onboard aircraft. The data is sent with Messages are divided in three classes: Telemetry, Ground and Datalink. GCS sends Ground data to Aircraft. Server receives Telemetry raw data, dispatches them and send them to GCS via Ground data. The mission will be defined by Mission definition module and then sent to Planner. The Planner interconnects to GCS for following aircrafts operations as interaction and monitoring level. Noting that IVY is an open-source software bus that can plug modules from many different languages such as C, C++, java, python, etc. Thus we can add and modify easily the new modules in paparazzi platform.

IV. SCENARIO

The first scenario we did for integrating the mission planner module in Paparazzi is shown in Figure 2. The scenario explains a common problem of configuration system in UAV domain. Multiple UAVs would cooperate together to solve separate tasks. The mission is composed of the surveillance task in a two-sections zone. For this mission, we have three UAVs (of the same type - fixed wing aircraft for the first simulation) and one operator. Basically, the main task is the surveillance of a large section with several UAVs. This task is decomposed to the surveillance task in two subsections each for only one UAV. The reconfiguration phase happens when

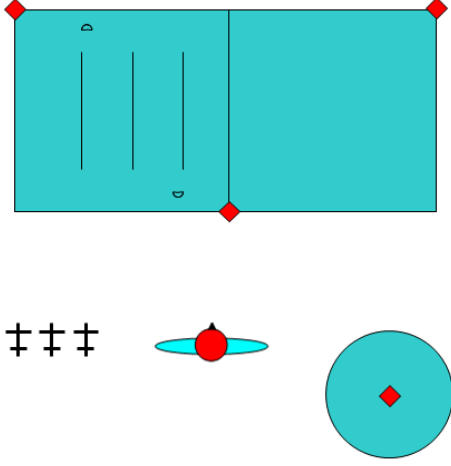


Fig. 2. The scenario with 3 aircrafts, 2 tasks and one operator.

an UAV is lacking of battery, the backup UAV will replace this one. The background tasks include then the separation phase - allocation of task and the deconflict phase - collision avoidance.

The goal of the mission planner is to allocate the two tasks to the three aircrafts. As the result (see Figure 3 (a) and (b) for more details), two tasks are allocated to two aircrafts. The other aircraft is circling outside around its standby point.

V. RESULTS

The simulation result is shown by the video <https://www.dropbox.com/s/gy9djtkyrtqdb7/video3AC1.avi> in using Paparazzi simulation platform. Two surveillance sections are defined by S1-S2 and S2-S3 rectangle. We have three standby points for these three aircrafts. The HOME point is the point of starting our aircrafts and the other points are used for taking-off and landing. The mission planner will run as following steps:

- Launching three aircrafts: we launch our three aircrafts in blue, yellow and red (see Figure 4).
- Initializing planner: These three aircrafts are sent to their three standby points STDBY1, STDBY2, STDBY3. They are circling around these points and waiting for the next commands (see Figure 5).
- Starting planner: We are allocating two surveillance tasks to the three aircrafts (see Figure 6). Basically, they are distributed to the red and yellow aircraft because these two aircrafts are nearer from the two tasks than the blue aircraft. The trajectories of the surveillance task are defined by dividing the zone as the parallel lines. We will follow all these lines to cover all the zone. The radius of the circle once the aircraft goes out of the zone is determined in the flight plan before flying and can not be changed during flying (see Figure 7). At that moment, the status of two tasks is changed to ALLOCATED and the status of the red and yellow aircraft is changed to WORKING.
- Executing and replanning: Three aircrafts are continuing doing their current tasks. The planner is detecting

any problem during flying including the battery, connection, etc. We simulate here a problem concerning the battery of the red aircraft by adding a “Kill AC” button. The red aircraft is supposed to land and we considered it to be dead (see Figure 8). The status of this aircraft is changed to DEAD. The planner detects the problem and make an online replanning. As the result, the blue aircraft will be sent to replace the red one (see Figure 9) (and the status of the blue aircraft is changed to WORKING after on). It is continuing the surveillance task of the red aircraft. We have now two working aircrafts are doing two tasks and one aircraft is dead (or preparing) outside. If we come to activate this aircraft again, we will send it to its standby point and waiting for the substitution.

On the computer Dell Latitude E6520 (Intel(R) Core(TM) i7-2640M CPU @ 2.80GHz, Ubuntu 12), the computation time measured during the simulation above are as following:

| Initialization phase | Starting phase | Executing phase | Updating phase |
|----------------------|----------------|-----------------|----------------|
| 120 (ms) | 326 (ms) | 97 (ms) | 305 (ms) |

We found that the computation time depends on the complexity of the mission such as the number of tasks, the number of aircrafts, the environment and the type of tasks, etc.

VI. CONCLUSION

The paper presents a new mission planner module integrated in Paparazzi platform. This planner allows us to allocate tasks to aircrafts in the context of cooperation between multiple UAVs. With this planner, we could carry out projects of surveillance tasks by Paparazzi. As the preliminary results of this paper, we perform a scenario of three aircrafts with the same type (fixed-wing aircraft). The simulation shows that we can deal with the replacement/substitution of multiple UAVs when there are problems of battery, connection missing, etc. The modality of the mission planner is currently based on the centralized communication system. For the next step of this work, we will simulate with other scenarios for different tasks, different types of aircrafts and use the ad-hoc network for the distributed communication system as well. We are going to apply this Mission planner paparazzi module in the D3CoS project to make a demonstrator of UAV domain. In the demonstrator, we will integrate not only our mission planner for the cooperation issue, but also the other tools to guarantee the human-machine interaction. The demonstration with real aircrafts will be performed at the end of this year.

ACKNOWLEDGMENTS

This work is financed by the european project “D3CoS”.

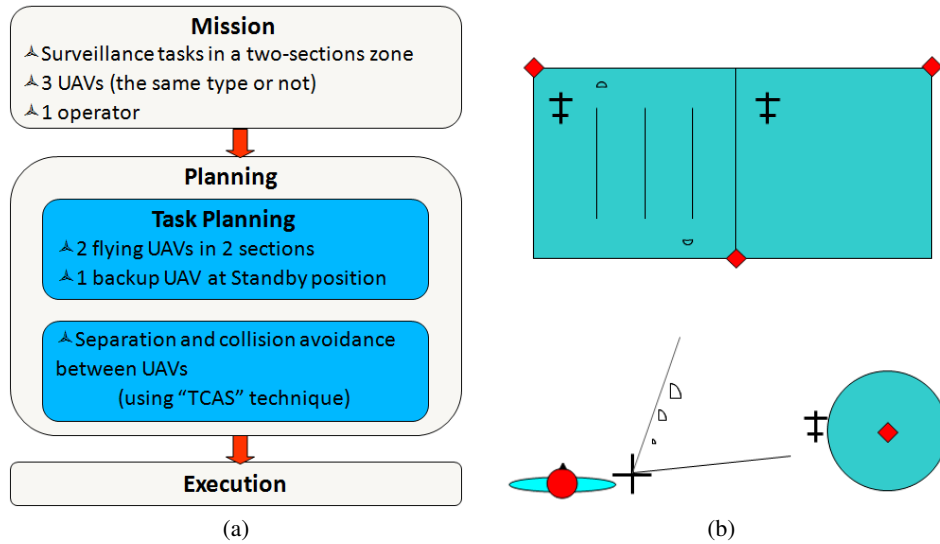


Fig. 3. The mission planner process (a) and the distribution of two tasks to three aircrafts (b).

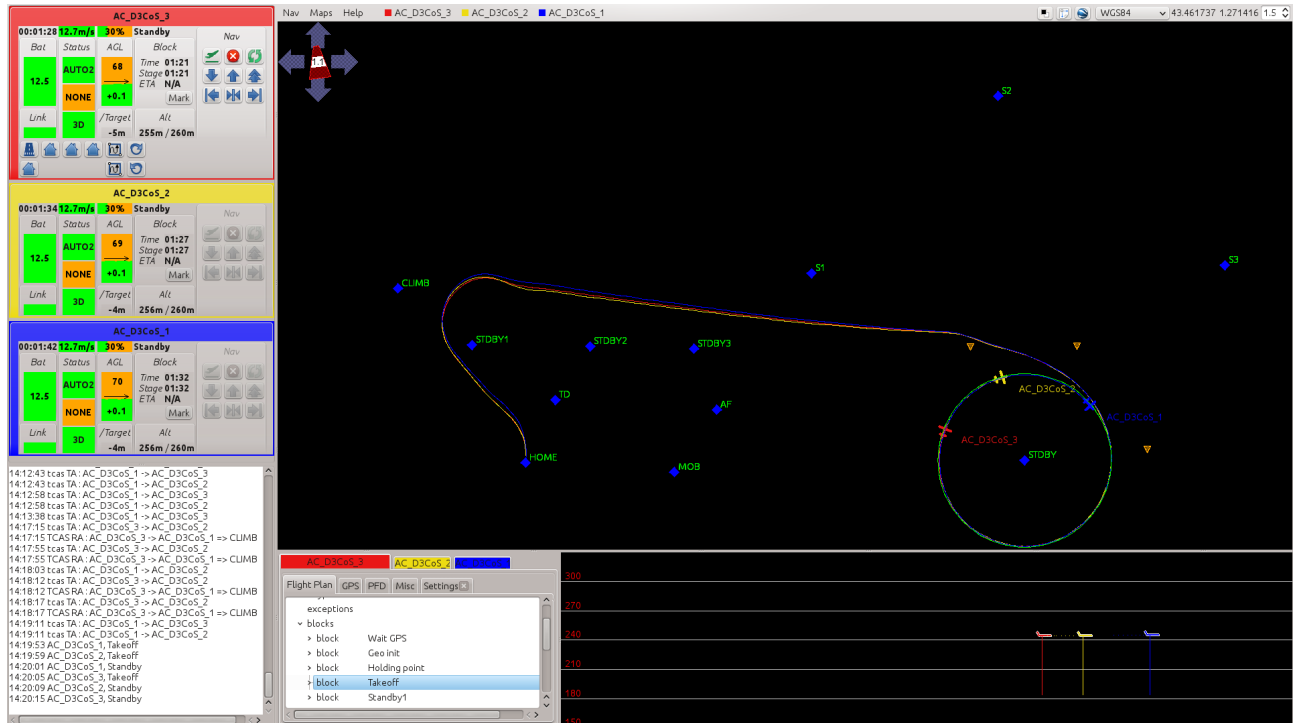


Fig. 4. Launching stage: three aircrafts in red, blue and yellow at the HOME point at the beginning. All the waypoints are in the UTM local coordinate relative to the the HOME point. We can see in the upper left corner the GCS interface of Paparazzi platform with strips of button for controlling/monitoring/editing before and during flight. The warning messages system is found on the lower left corner. The blocks of tasks and the altitude monitoring interface are on the lower side of the figure. At the end of the phase, three aircrafts are circling around the STDBY point.

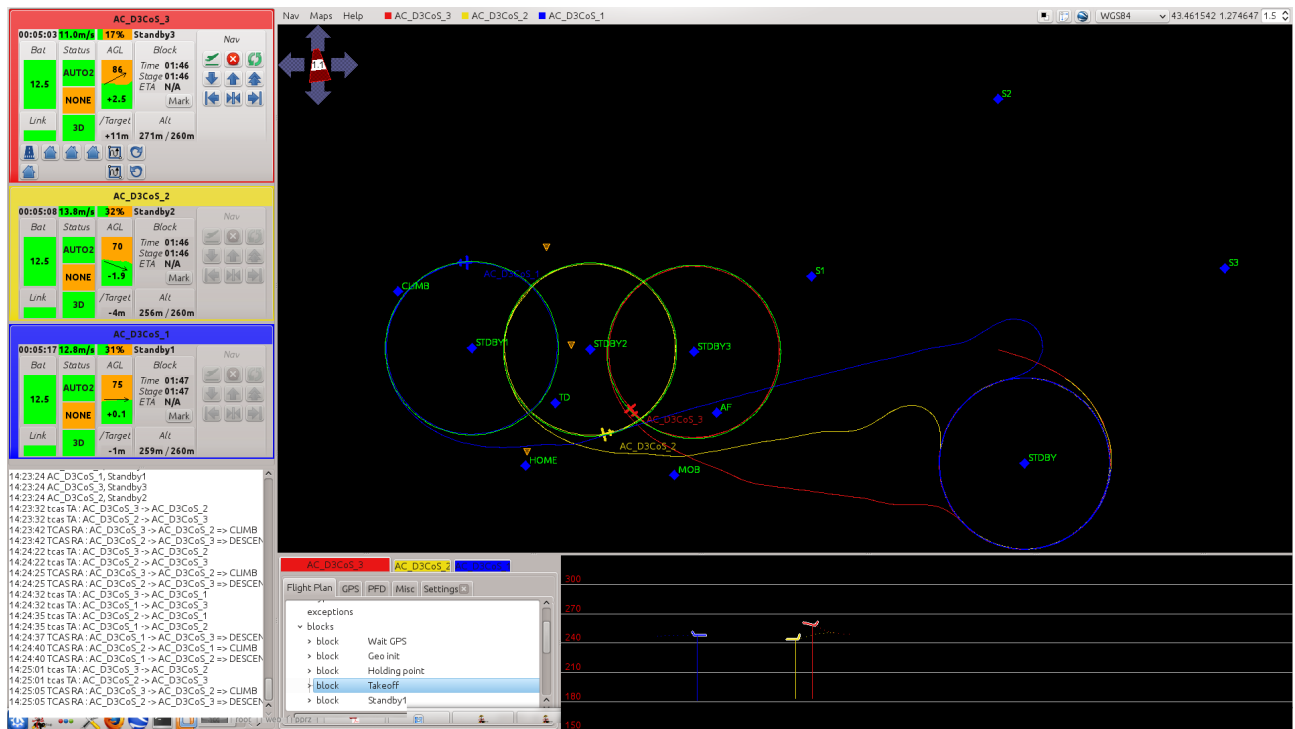


Fig. 5. Planner initialization: 3 aircrafts are circling around their standby points STDBY1, STDBY2, STDBY3 and waiting for the next commands. During flight, Collision avoidance module (using "TCAS") is always triggered each 500 (ms) and the conflict is detected if it happens. In this figure, we can see the conflict between the red and yellow aircraft. As the result, the red aircraft is climbing and the yellow one is descending for avoiding the collision between them. We can see this phase in the altitude interface on the lower side of the figure.

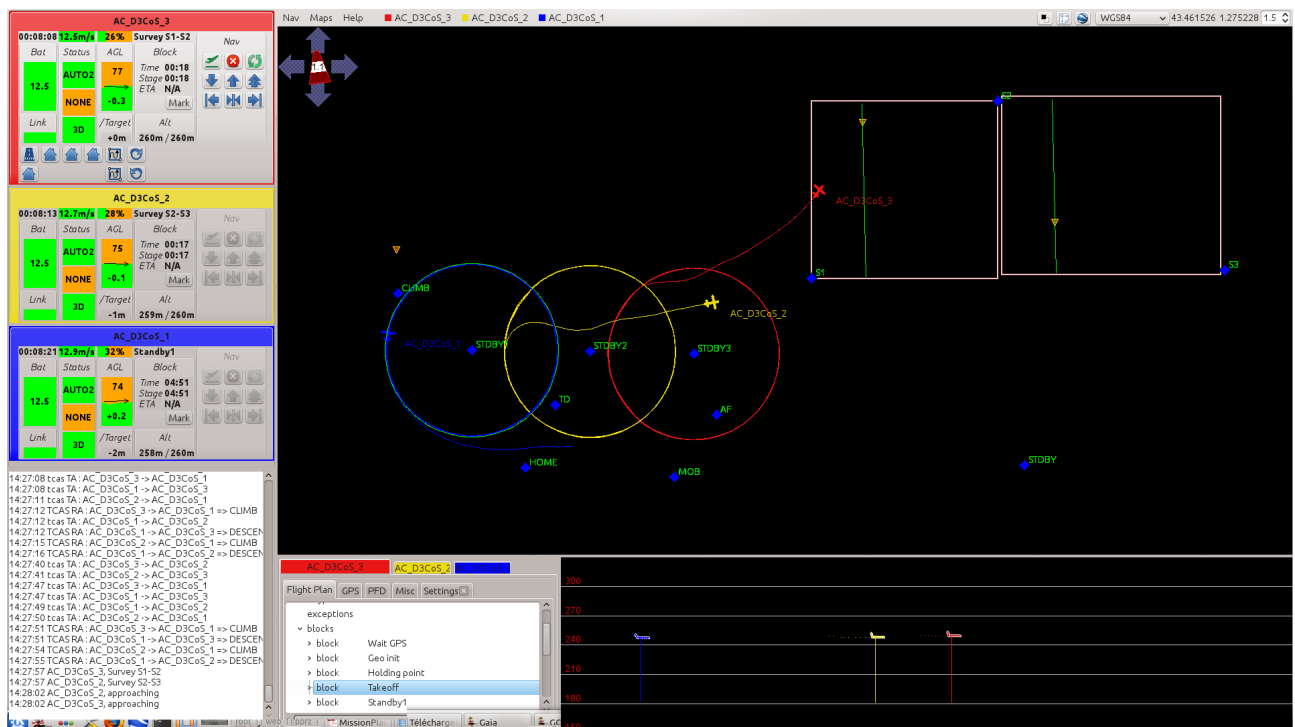


Fig. 6. Start planner: the planner calculates and minimize the cost of the insertion of each task to each aircraft. Then, it chooses the yellow and the red aircrafts to two tasks of surveillance because they are nearest from the two tasks than the blue one.

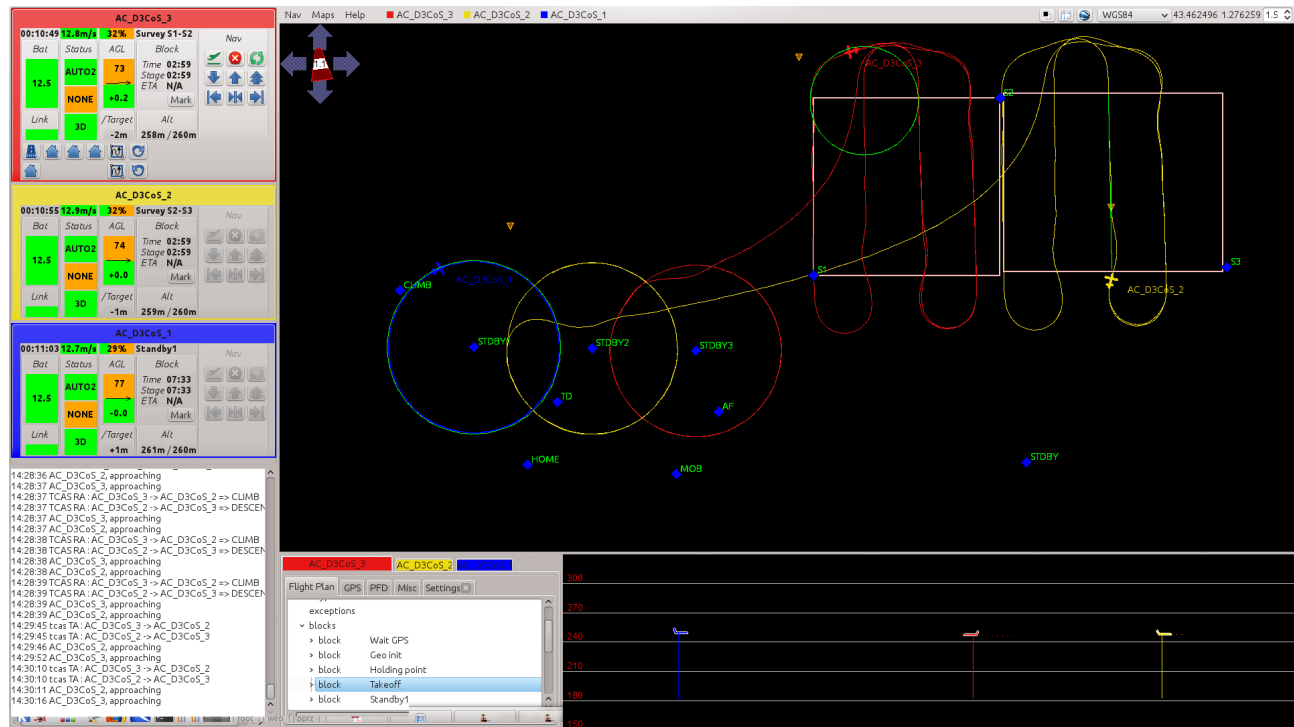


Fig. 7. Start planner: two aircrafts are doing their surveillance tasks in two sections. We can see here the trajectory shapes during the surveillance task

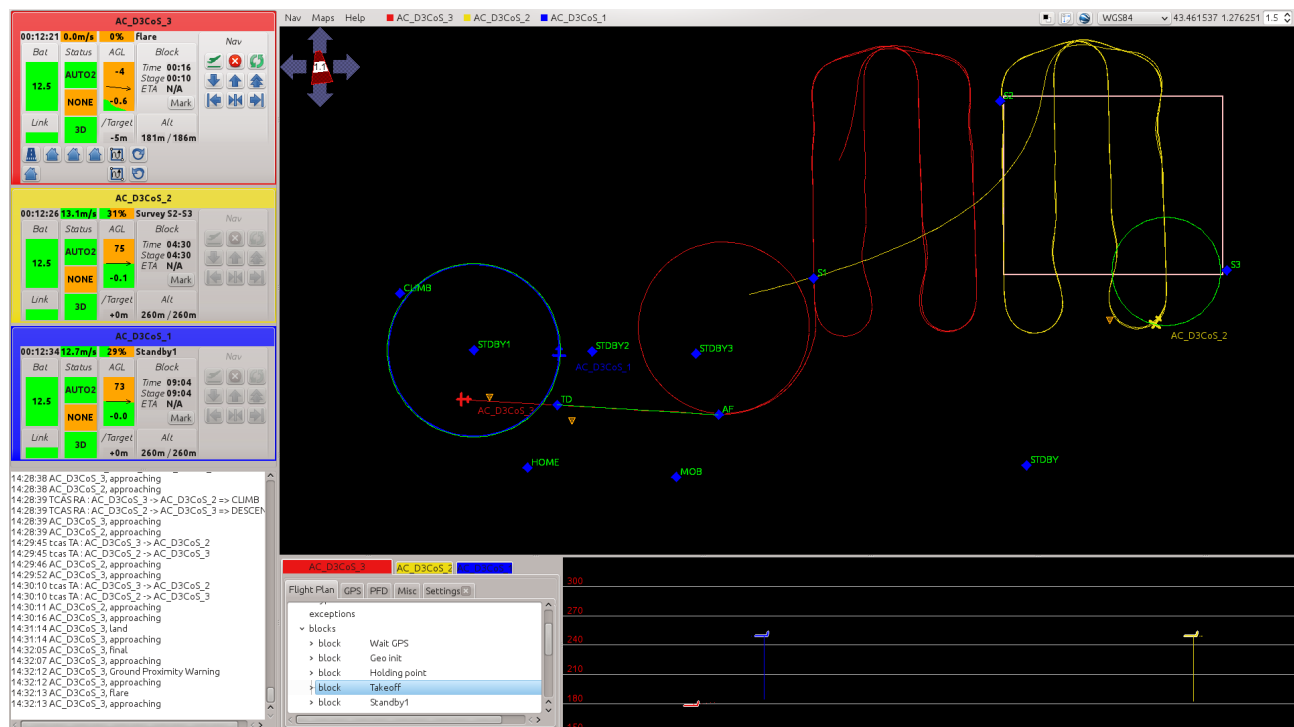


Fig. 8. Kill aircraft simulation: end of landing after doing all the landing procedure through AF and TD points

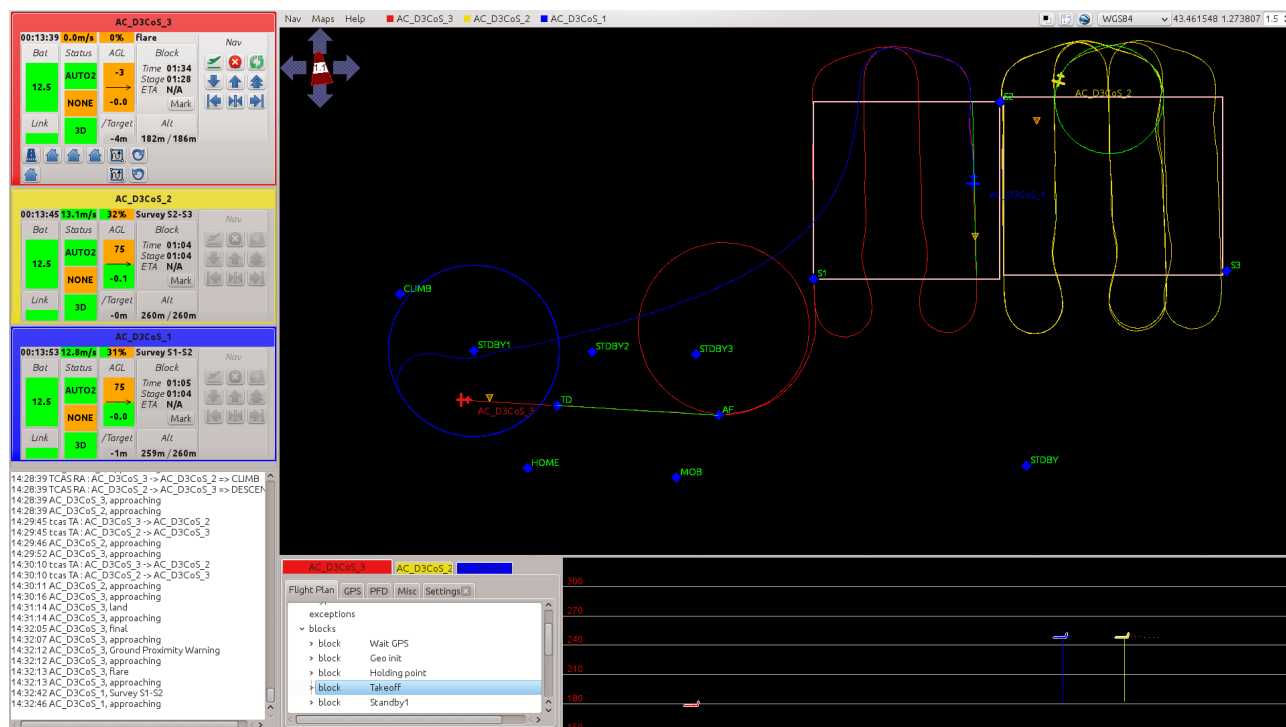


Fig. 9. Updating planner: recalculating the cost of insertion and choosing the available blue aircraft to replace the red one for doing the S1-S2 section surveillance task

REFERENCES

- [1] Brisset, P. and Drouin, A. and Gorraz, M. and Huard, P.S. and Tyler, J., *The paparazzi solution*, MAV2006, Florida, 2006.
- [2] Brisset, P. and Hattenberger, G., *Multi-UAV Control with the Paparazzi System*, Proceedings of the First Conference on Humans Operating Unmanned Systems (HUMOUS08), Brest, France, 2008.
- [3] Bellingham, J., *Multi-task allocation and path planning for cooperating UAVs*, Citeseer, 2001.
- [4] Chantry, E. and Barbier, M. and Farges, J.L. and Center-FRANCE, O.T., *Planning algorithms for autonomous aerial vehicle*, 16th IFAC World Congress, Prague, 2005.
- [5] Gancet, J. and Hattenberger, G. and Alami, R. and Lacroix, S., *Task planning and control for a multi-UAV system: architecture and algorithms*, IEEE/RSJ International Conference on Intelligent Robots and Systems, Canada, 2005.
- [6] Gerkey, B.P. and Mataric, M.J., *A formal analysis and taxonomy of task allocation in multi-robot systems*, The International Journal of Robotics Research, 2004.
- [7] Grøtli, E.I. and Johansen, T.A., *Task assignment for cooperating UAVs under radio propagation path loss constraints*, Journal of Intelligent & Robotic Systems, Springer, 2012.
- [8] Lagoudakis, M.G. and Berhault, M. and Koenig, S. and Keskinocak, P. and Kleywegt, A.J., *Simple auctions with performance guarantees for multi-robot task allocation*, IEEE/RSJ International Conference on Intelligent Robots and Systems, Japan, 2004.
- [9] Mosteo, A.R. and Montano, L. and Lagoudakis, M.G., *Multi-robot routing under limited communication range*, IEEE International Conference on Robotics and Automation (ICRA 2008), California, USA, 2008.
- [10] Pongpunwattana, A. and Rysdyk, R., *Real-time planning for multiple autonomous vehicles in dynamic uncertain environments*, Journal of Aerospace Computing, Information, and Communication, 2004.
- [11] <http://paparazzi.enac.fr>, Paparazzi, Open-source hardware and software project.
- [12] <http://www.tls.cena.fr/products/ivy>, Ivy, A simple protocol and a set of open-source (LGPL) libraries.