



**HAL**  
open science

# The Task Agent Resource Function application in UAV domain

Tan-Viet-Anh Truong, Petr Benda, Jiri Vokrinek

► **To cite this version:**

Tan-Viet-Anh Truong, Petr Benda, Jiri Vokrinek. The Task Agent Resource Function application in UAV domain. AIAA GNC 2013, AIAA Guidance, Navigation and Control Conference, Aug 2013, Boston, United States. pp xxxx, 10.2514/6.2013-4798 . hal-00936079

**HAL Id: hal-00936079**

**<https://enac.hal.science/hal-00936079>**

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 Task Agent Resource Function application in UAV domain

Tan Viet Anh Truong \*  
*Ecole National Aviation Civil (ENAC)*

Petr Benda<sup>†</sup> and Jiri Vokrinek\*  
*Czech Technical University (CTU)*

The goal of the paper is to present an application of TARF (Task Agent Resource Function) in UAV domain. This TARF is used to optimize the tasks allocation in case of cooperation between multiple Unmanned Aerial Vehicles. It plays the role of a generic mission planner for cross domain such as UAV, maritime, automotive and manned aerial vehicle (MAV). This paper concentrates on how to integrate TARF in Paparazzi platform – an UAV platform based on PPRZ autopilot – in order to perform a swarm of aircrafts doing different tasks. The TARF contributes a new module to this platform for the simulation and real demonstrations.

## I. Introduction

The Unmanned Aerial Vehicles (UAVs) have been developed more and more since the last decade. They are used in varied applications such as surveillance missions, gas pipelines monitoring,<sup>1</sup> localization and mapping,<sup>2</sup> meteorology<sup>3</sup> and photography.<sup>4</sup> Several UAV open source autopilot projects developed recently<sup>5-7</sup> are enable to manage the navigation and monitoring before and during the flight. However, these autopilots are still far from controlling and optimizing the multi-tasks mission for an UAV swarm. This paper proposes a TARF solution for Paparazzi autopilot.<sup>8,9</sup> 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.

The task allocation issue has been studied longtime ago, and recently in UAV domain (see<sup>10</sup> and<sup>11</sup> for examples). They focused not only on trajectory planning for agents<sup>12</sup> but also on how to distribute tasks for them.<sup>13</sup> The sequence of task is given at the desired time for one agent<sup>14</sup> and more complicated for multi agents.<sup>15</sup> Recently, the online task planning<sup>16,17</sup> is also conducted after dealing with the offline one.<sup>18</sup> The goal of this paper is to use the TARF as an online mission planner for multi-UAVs. This mission planner is then integrated in Paparazzi and contributing a new module to this platform. The Paparazzi community could perform different kinds of the cooperation of multiple UAVs afterward due to this module.

Among different problems of cooperation of multiple UAVs, the mission planning is merged as the main activity. The principal question is 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 describe the mission planner TARF developed by ENAC (Ecole National Aviation Civil) and CTU (Czech Technical University). 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 simulations were done by using the Paparazzi simulation platform. In the first simulation, we dealt with the scenario of three aircrafts doing three surveillance tasks.

The paper is organized as following. We first describe our TARF 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.

---

\*Researcher, UAV team, ENAC, 7 Avenue douard Belin, 31055 Toulouse Cedex 04, France, tan-viet-anh.truong@enac.fr

<sup>†</sup>Researcher, Czech Technical University, Prague, 16017, Czech, petr.benda,jiri.vokrinek@agents.fel.cvut.cz

## II. The TARF integration in Paparazzi

### II.A. TARF description

The TARF algorithm (see in Figure 1) based on the multi-agents solver that minimizes the insertion cost computed by resource agents.<sup>19</sup>

---

**Algorithm 1** The abstract algorithm of a multi-agent solver.

Input : Set of tasks  $T$ , set of Resource Agents  $R$

Output :  $T$  allocated on  $R$   
and local plans of Resource Agents exist

```
function solve( $T, R$ ) begin
  apply ordering heuristic on  $T$ 
  forall  $t : T$  begin
    allocateCNP( $t, R$ )
    if allocation not successful then
      exit with failure or
      mark  $t$  as not allocated and continue
    end
    forall  $a : R$  begin
      apply dynamic improvement strategy
    end
  end
  forall  $a : R$  begin
    apply final improvement strategy
  end
end

function allocateCNP( $t, R$ ) begin
  forall  $a : R$  begin
    find winner with the lowest insertion
    estimation of  $t$ 
  end
  if winner is found then
    assign  $t$  to the winner
  else
    allocation not successful
  end
end
```

---

Figure 1. The TARF algorithm of task allocation,<sup>19</sup> where  $a$  and  $t$  are agent and task, respectively and CNP is contract net protocol.<sup>20</sup>

### II.B. Mission formulation and TARF API (Application Programming Interface)

We define the mission as the following data structures:

Mission = {Waypoint, Task, Aircraft}

Aircraft = {Aircraft Type, Aircraft Status, Speed}

Task = {Task Type, Task Position, Task Status, Duration}

Waypoint = {x, y, altitude}

TARF connects to other modules via the API as in Figure 2. The input of TARF (TARFin) is all the information about the mission including agents (aircrafts), tasks and other resources. The output of TARF (TARFout) is the list of agents (aircrafts) and the corresponding (assigned) tasks of each aircraft.

### II.C. TARF–Paparazzi google protocol buffer

Because of the different programming languages used in TARF (in java language) and Paparazzi at ground station level (in c language), a TARF–Paparazzi protocol buffer is proposed as in Figure 3. This protocol buffer developed by Google uses the server/client mechanism. The TARF client (in Paparazzi platform) sends information about the tasks and aircrafts to TARF server (Task allocator) and receives the allocation update including allocated tasks. The “mission planner” in Paparazzi monitors and sends all the feedbacks during

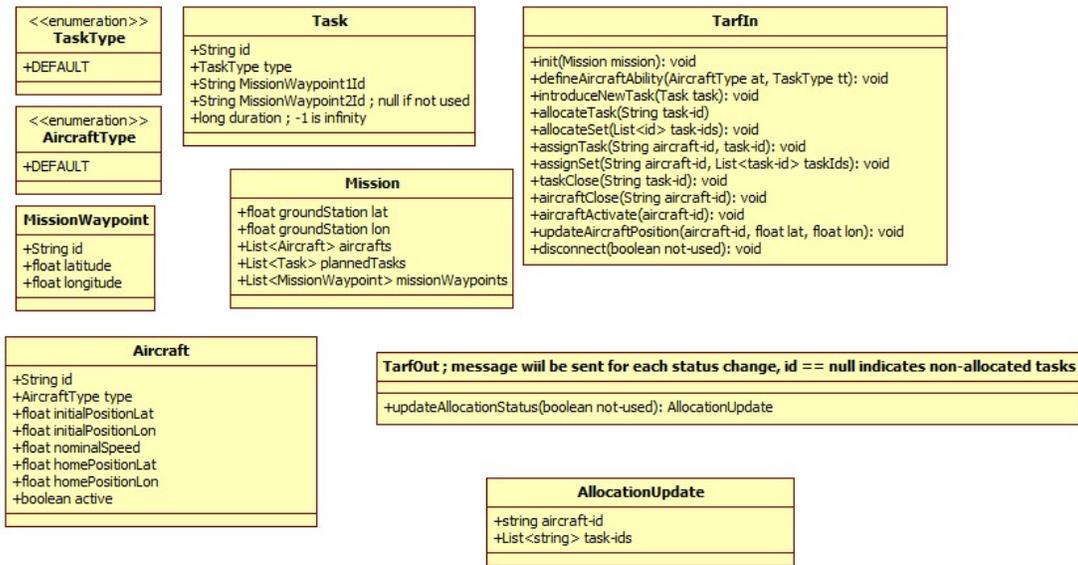


Figure 2. The API of TARF.

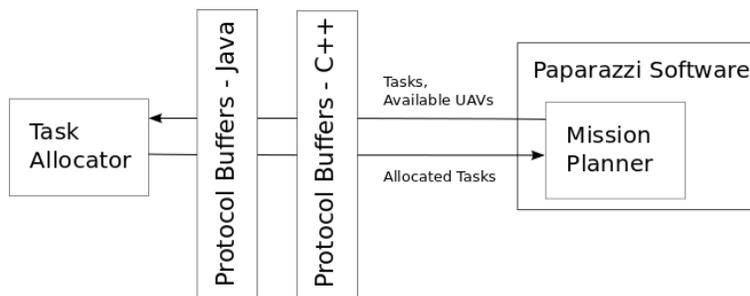


Figure 3. The architecture of TARF-Paparazzi google protocol buffer.

flight to TARF client. There is also the low level planning of collision avoidance in Paparazzi platform. When there is a conflict between two UAVs, they will change their altitudes to guarantee their desired trajectories.

### III. UAV Demonstration architecture

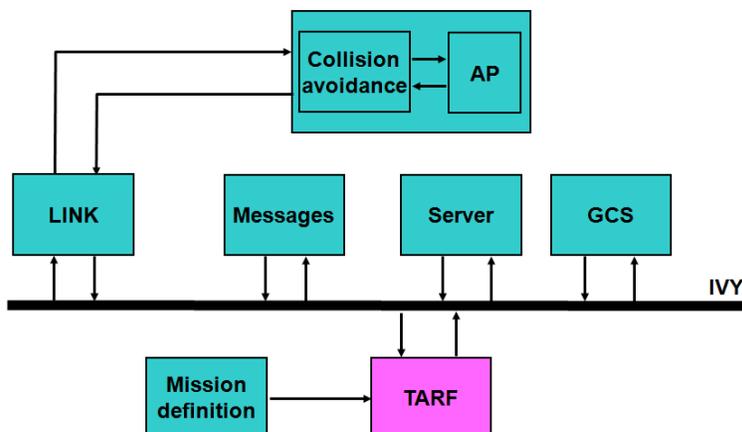


Figure 4. The simulation architecture based on Paparazzi platform.

The Figure 4 shows the demonstration architecture of one UAV in the cooperation between UAVs. This architecture consists of the Paparazzi modules in dark green and the new module TARF in pink.

Each aircraft is equipped with an autopilot (AP) which includes the navigation, stabilization during flight as well as a Collision Avoidance module. This collision avoidance use the “TCAS” technique to detect and avoid the collision between UAVs (see on the Paparazzi website<sup>9</sup> for more details about this module).

Aircrafts connect to the Ground Control Station (GCS) by the LINK agent. All the telemetry data and uplink (control) data are stored and dispatched by Server via a system of particular messages (in module Messages). All these agents are interconnected by IVY bus,<sup>21</sup> an open-source software bus that can plug modules from many different languages such as C, C++, java, python, etc.

The mission is defined by the operator in a XML file as an input of the TARF. TARF collects the data of tasks, agents and resources of the mission. Then it allocates these tasks to aircrafts by sending the commands to these aircrafts via IVY. Paparazzi ground segment monitors the flight and send the feedbacks to TARF.

### IV. Scenario

The scenario (in Figure 5) shows a typical issue in case of multi-UAVs. When an aircraft in an UAV swarm is lacking of battery, all the tasks will be redistributed to the rest aircrafts. The mission is composed of the surveillance task in a three different zones. For this mission, we have three UAVs (of the same type – fixed wing aircraft for this simulation) and one operator. Basically, the main task is the surveillance of the large section with several UAVs. This task is decomposed to the surveillance tasks in three different sections each for only one UAV. The reconfiguration phase happens when an UAV is lacking of battery, one of the other UAVs will replace this one to finish its on-going task. The background tasks include then the separation phase – allocation of task and the deconflict phase – collision avoidance.

### V. Results

The simulation is done by using Paparazzi platform and results are shown by the following video:

<https://www.dropbox.com/s/y257cezyajoca8/demotarf.mp4?n=141175626>

Three surveillance sections are respectively defined by  $(200 \times 350(m))$  S1-S2,  $(220 \times 390(m))$  S2-S3 and  $(200 \times 370(m))$  S4-S5 rectangle. The flight zone in the simulation is taken from the Muret (a small region near Toulouse, France) where our real demonstration will be carried out afterward. The google earth map of this region is loaded and served in all the figures of simulation as the background of GCS. All three aircrafts

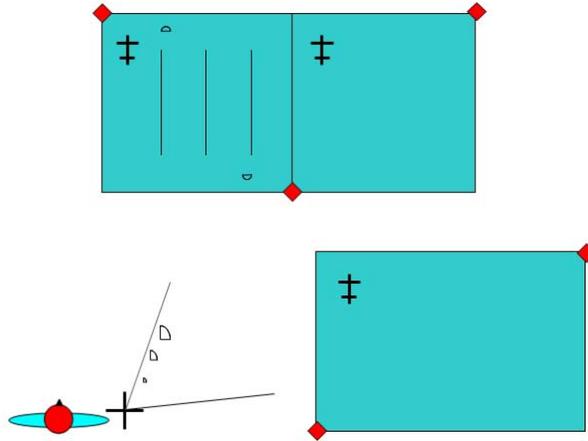


Figure 5. The scenario with three aircrafts, three tasks and one operator.

in the simulation use the Funjet aerodynamic model and the typical fixed-wing aircraft airframe and setting parameters. They are launched from the HOME waypoint and reaching first their standby points before doing their tasks. The other waypoints such as CLIMB, AF, etc are used for the the take-off and landing procedure. The TARF planner is running as following steps:

- Launching three aircrafts: They are launched from the HOME point one by one (see Figure 6).
- Initializing TARF: These three aircrafts are sent to their three standby points STDBY1, STDBY2, STDBY3 after taking-off. Normally, they are circling around these points and waiting for the next commands. In the simulation, all these procedures are reduced and the initiation step begins when the TARF–Paparazzi interface is called (see Figure 7). In this stage, TARF read all the data from mission XML file and parse them to the particular messages of waypoints, tasks, aircrafts.
- Starting TARF: TARF allocates three surveillance tasks to the three aircrafts (see Figure 8). Basically, TARF distributes respectively tasks to available agents. At the beginning, three aircrafts are available and the first task (S1-S2 surveillance task) is allocated to its nearest aircraft – the aircraft AC\_D3CoS\_3 in red. The second task (S2-S3 surveillance task) is then allocated to the aircraft AC\_D3CoS\_2 in yellow and the third task (S4-S5 surveillance task) is allocated to the aircraft AC\_D3CoS\_1 in blue, respectively. During the flight, Collision Avoidance module is always triggered each 500 (ms) in order to detect the conflict between UAVs (see Figure 9 for an example). The trajectories of the surveillance tasks is tracked as in the Figure 10. These trajectories are defined by dividing the zone as the parallel lines. Aircrafts follow all these lines to cover their surveillance zones. The radius of the circle used for returning aircrafts once they goes out of the zone is determined in the flight plan before flying and can not be changed during the flight.
- Updating TARF: Three aircrafts are continuing doing their current tasks. One agent of GCS is detecting any problem during flying including the battery, connection, etc and send to TARF. The “Kill AC” button in the TARF–Paparazzi interface is used to simulate a problem concerning the battery of the aircraft AC\_D3CoS\_2 (see Figure 11). The yellow aircraft is then supposed to land and being considered as dead (Figure 12). The TARF receives the feedback of on-doing tasks, recalculates and sends the allocation update including the list of new tasks corresponding to aircrafts. In this case, the S2-S3 surveillance task is still not done and the two rest aircrafts (AC\_D3CoS\_1 and AC\_D3CoS\_3) are continuing to finish their own tasks (see Figure 13). This task is always waiting for one of two these aircrafts once it accomplishes its assigned task. Actually, the surveillance task is a cycle task. Aircrafts cover the zone as much as possible to obtain the goal which might be detecting targets or measuring the atmosphere inside the zone, etc. Our mission XML defines the duration for each surveillance task so that the aircraft could finish its task and go to another task if any. In our simulation, the durations of task 1 (S1-S2 surveillance task) and task 3 (S4-S5 surveillance task) are 400(s) and 700(s), respectively. As the result, TARF allocates the task 2 (S2-S3 surveillance task) to aircraft AC\_D3CoS\_3’s

list of tasks. Once aircraft AC\_D3CoS\_3 terminates task 1, it goes to task 2 and continue this task of AC\_D3CoS\_2 at the beginning (see Figure 14). These two aircrafts are doing their tasks until finishing the tasks durations (see Figure 15).

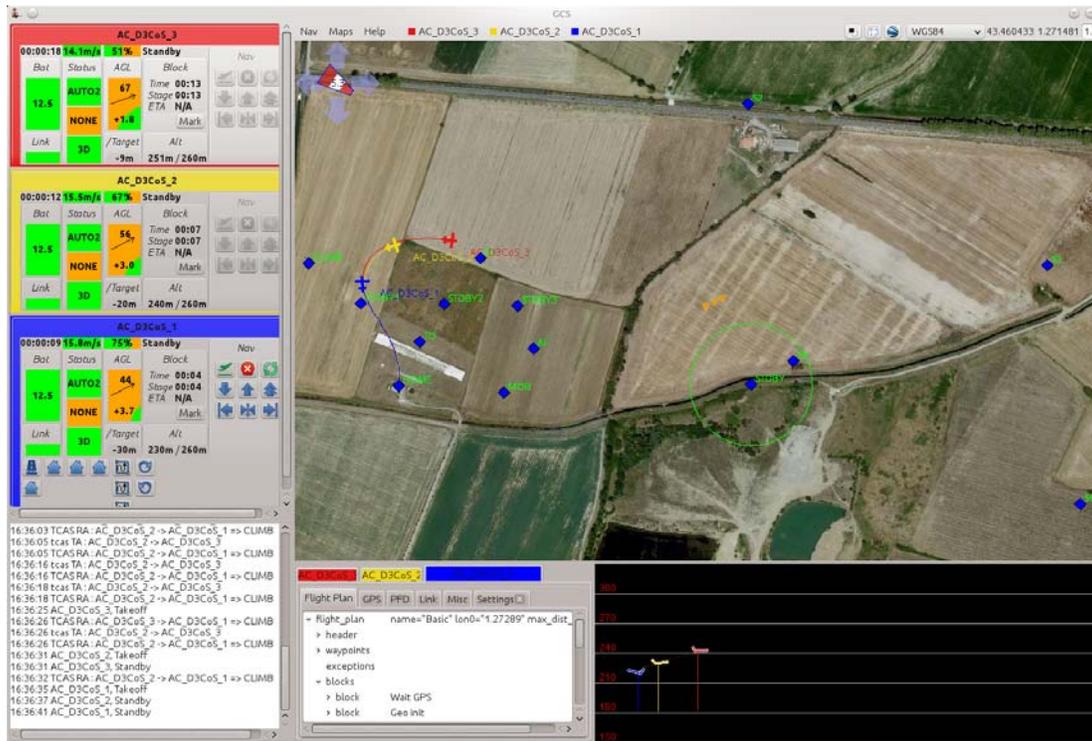


Figure 6. Launching stage: three aircrafts in red, blue and yellow are at the HOME point at the beginning. All the waypoints are in the UTM local coordinate relative to the the HOME point. We can see in the upper left corner the GCS interface of Paparazzi platform with strips of button for controlling/monitoring/editing 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.

## VI. Conclusion

The paper presents the TARF planner module integrated in Paparazzi platform. This planner allows us to allocate tasks to aircrafts in the context of cooperation between multiple UAVs. The simulation of the proposed scenario gave the good results. Aircrafts could cooperate together to perform the desired tasks set. The scenario in the paper deals with three aircrafts with the same type (fixed-wing aircraft). The replacement/substitution of multiple UAVs when there are problems of battery, connection missing, etc could be resolved with this TARF planner. For the next step of this work, we will simulate with other scenarios for the different tasks and the different types of aircrafts as well (that could be the combination between aircrafts and rotorcrafts). We are going to apply this TARF module within the D3CoS (Designing Dynamic Distributed Cooperative Human-Machine Systems)<sup>22</sup> project to make a real flight demonstrator of an UAV swarm. The TARF could also be used in other domains in the project such as automotive, manned aerial vehicle and maritime.

## Acknowledgments

This work is financed by the european project “D3CoS”. We thanks Gautier Hattenberger for his valuable help.

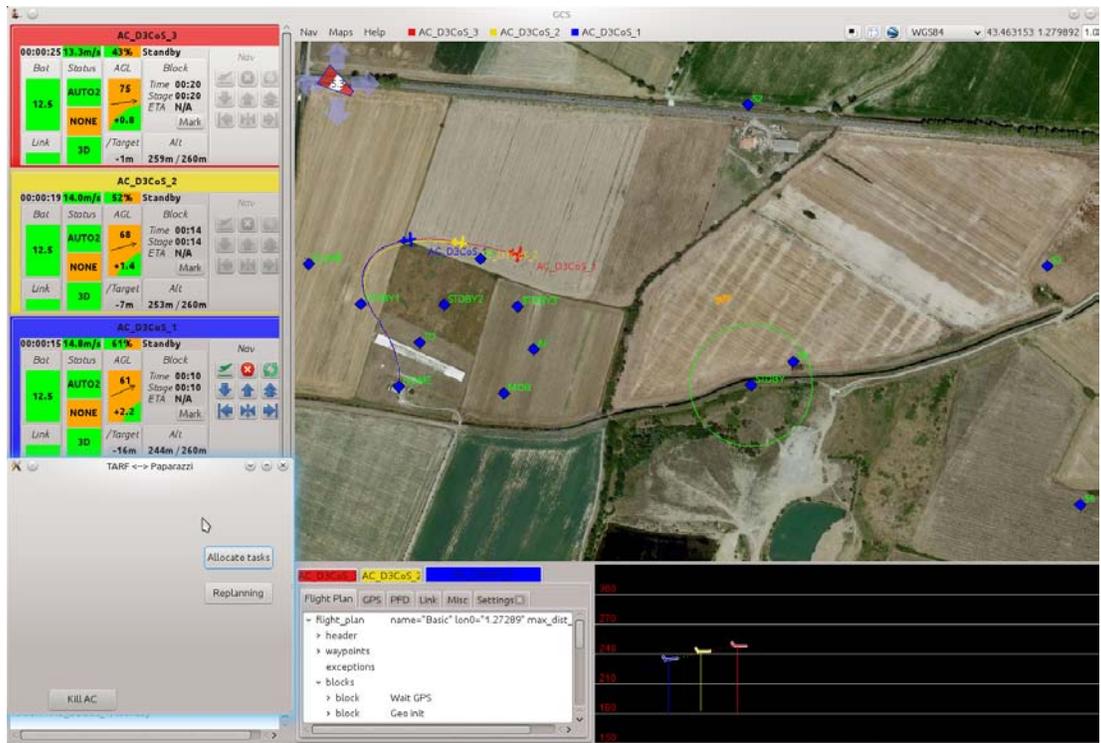


Figure 7. TARF initialization: three aircrafts are going their standby points STDBY1, STDBY2, STDBY3 and the TARF-Paparazzi interface is called on the left of the screen

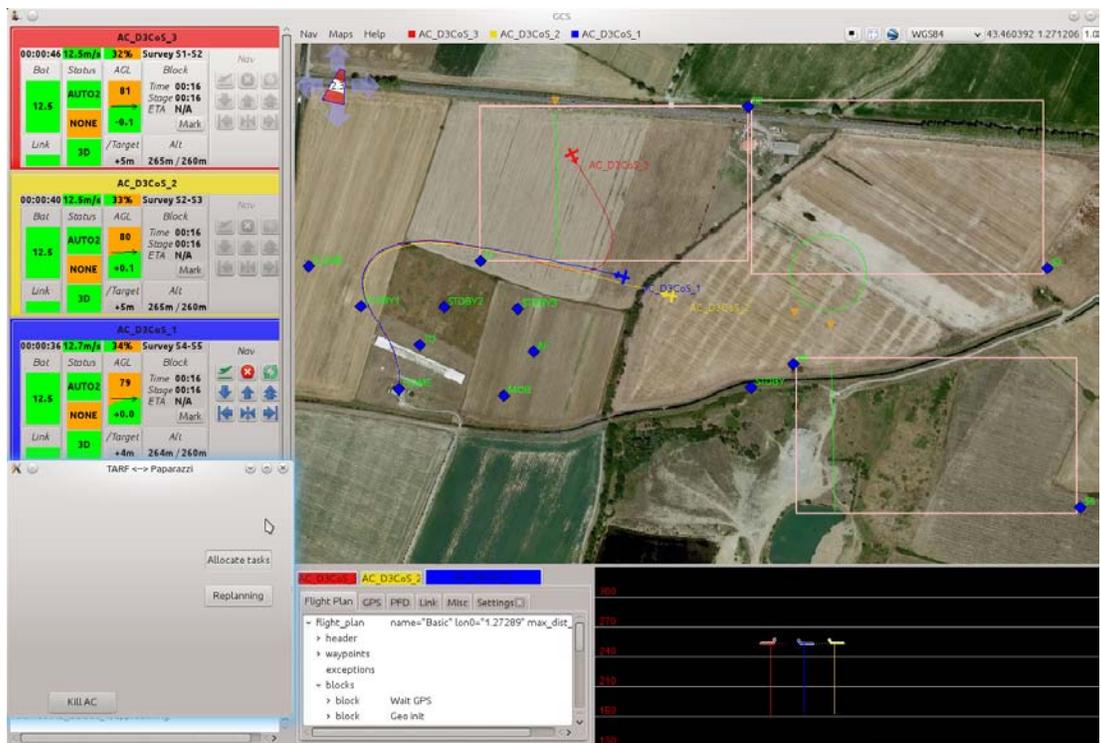


Figure 8. Start TARF: The TARF is started once the “Allocate tasks” button is clicked. It calculates and minimize the cost of the insertion of each task to each aircraft.

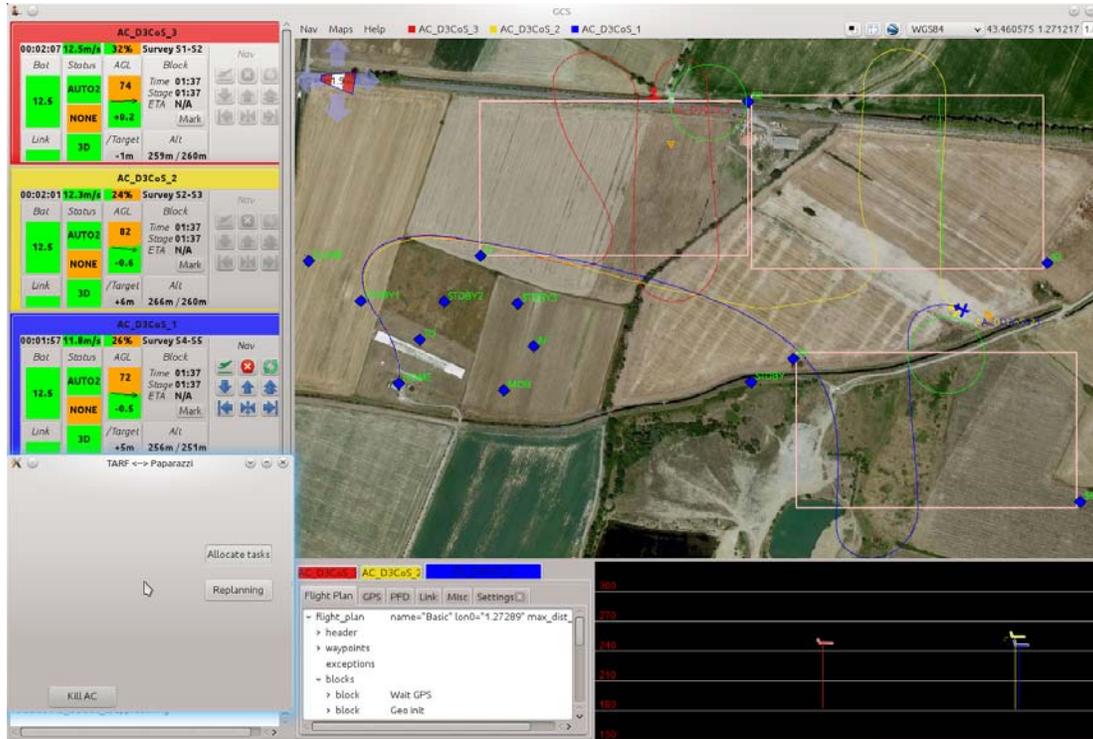


Figure 9. The conflict happens between the blue and yellow aircraft. As the result of using “TCAS” technique, the yellow aircraft is climbing and the blue one is descending for avoiding the collision between them. This phase of conflict is shown in the altitude interface on the lower side of the figure.

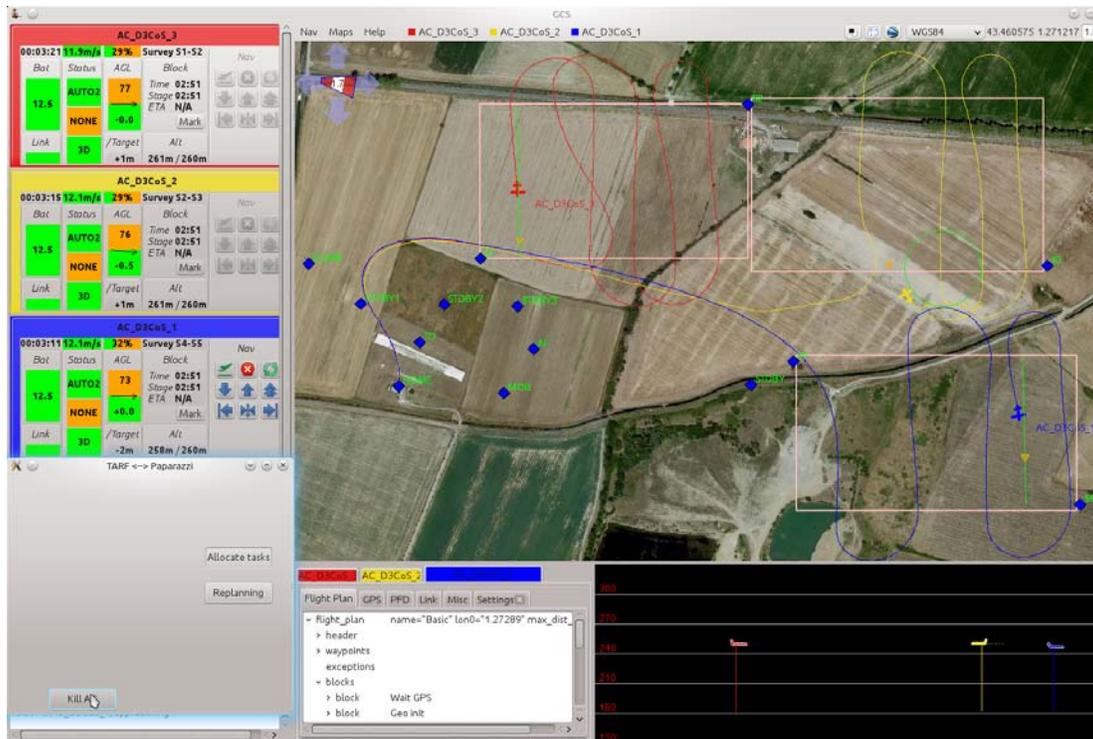


Figure 10. The trajectory shapes of three aircrafts doing their surveillance tasks. The grids is pre-calculated in autopilot

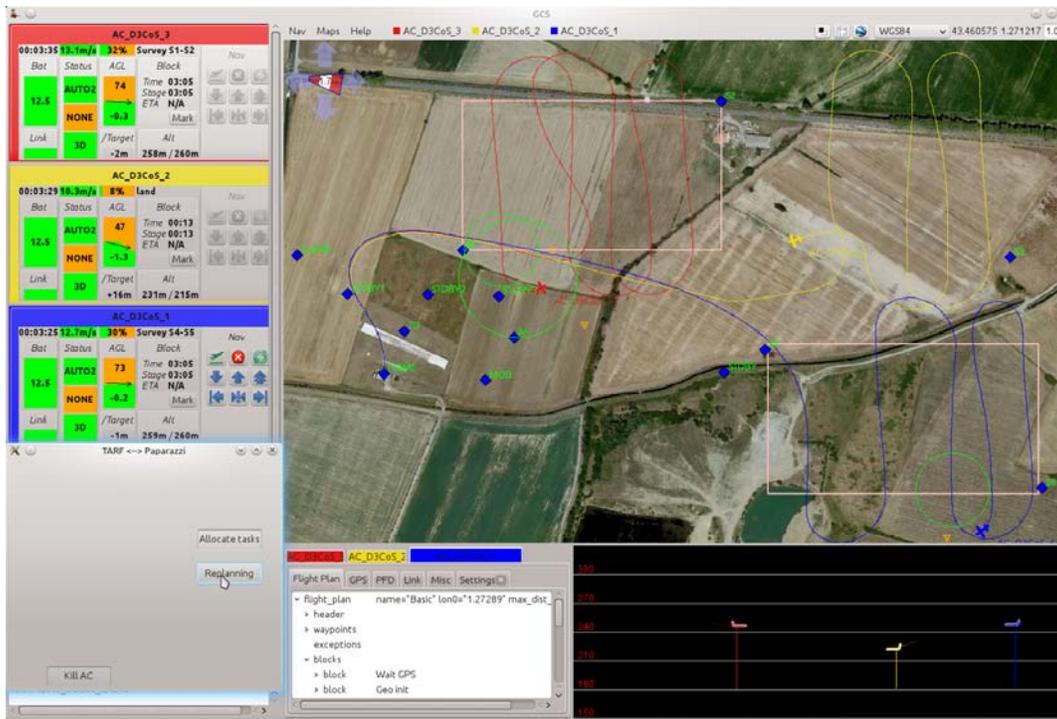


Figure 11. Kill aircraft simulation: at the beginning of the command, AC\_D3CoS\_2 starts to land

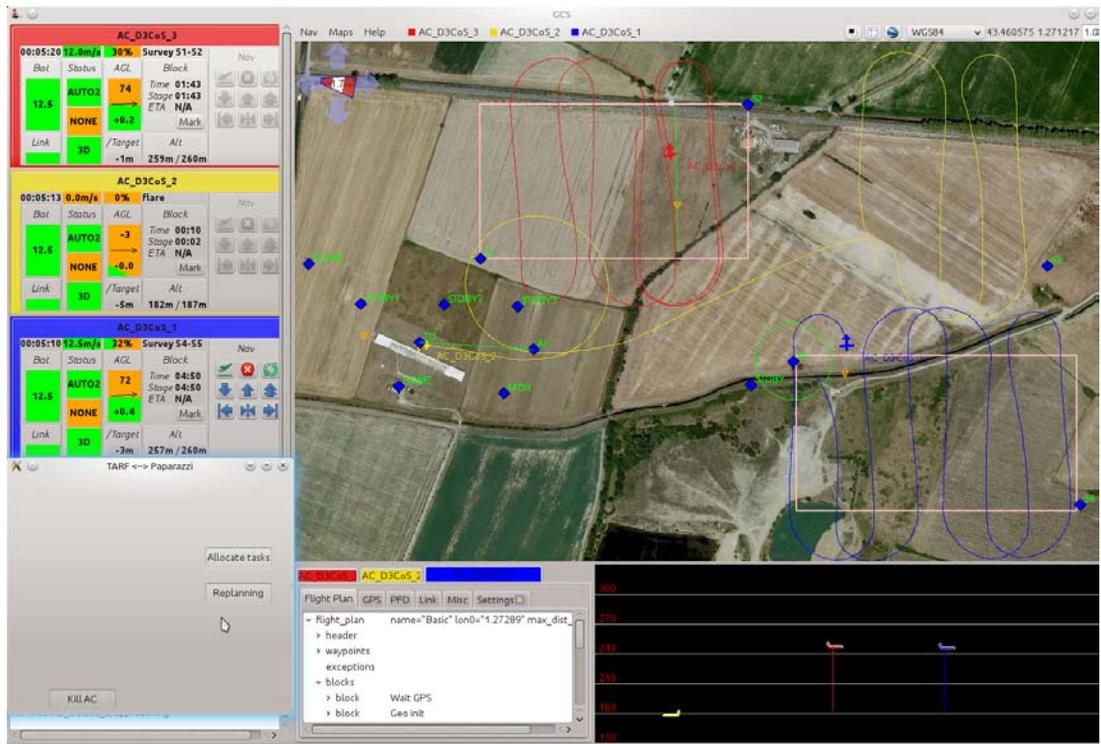


Figure 12. Kill aircraft simulation: end of landing after doing all the landing procedure through AF and TD points. The trajectory of landing procedure is tracked in yellow

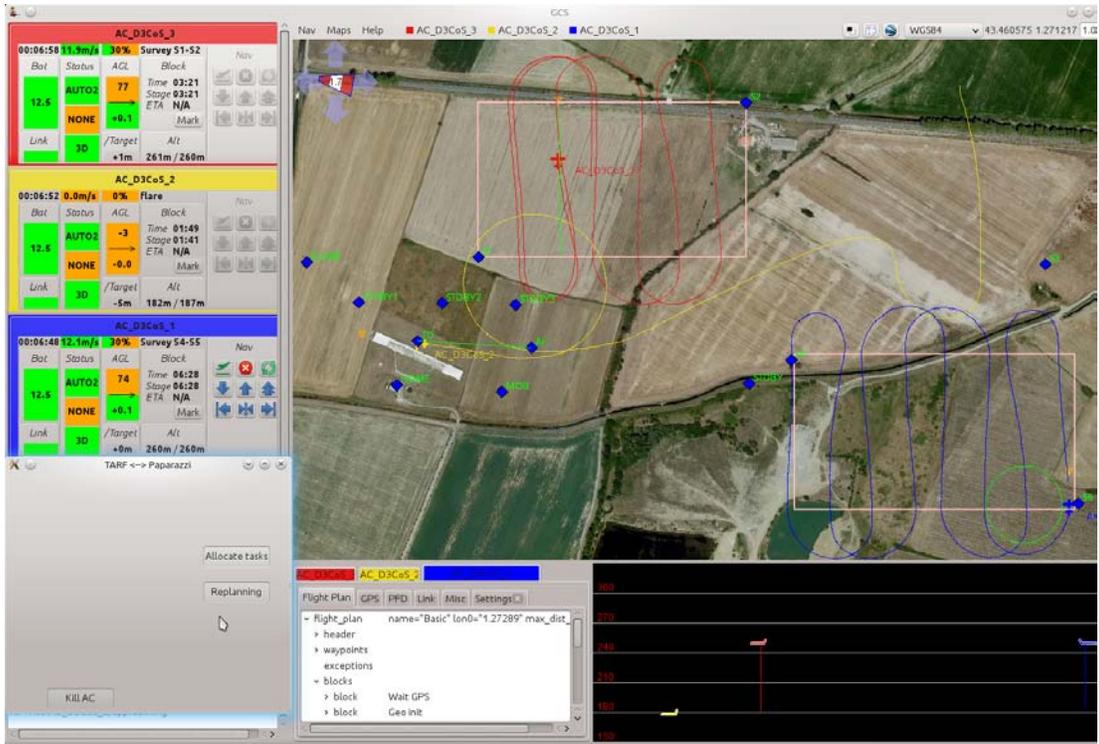


Figure 13. Aircrafts are continuing doing their surveillance tasks until finishing the durations

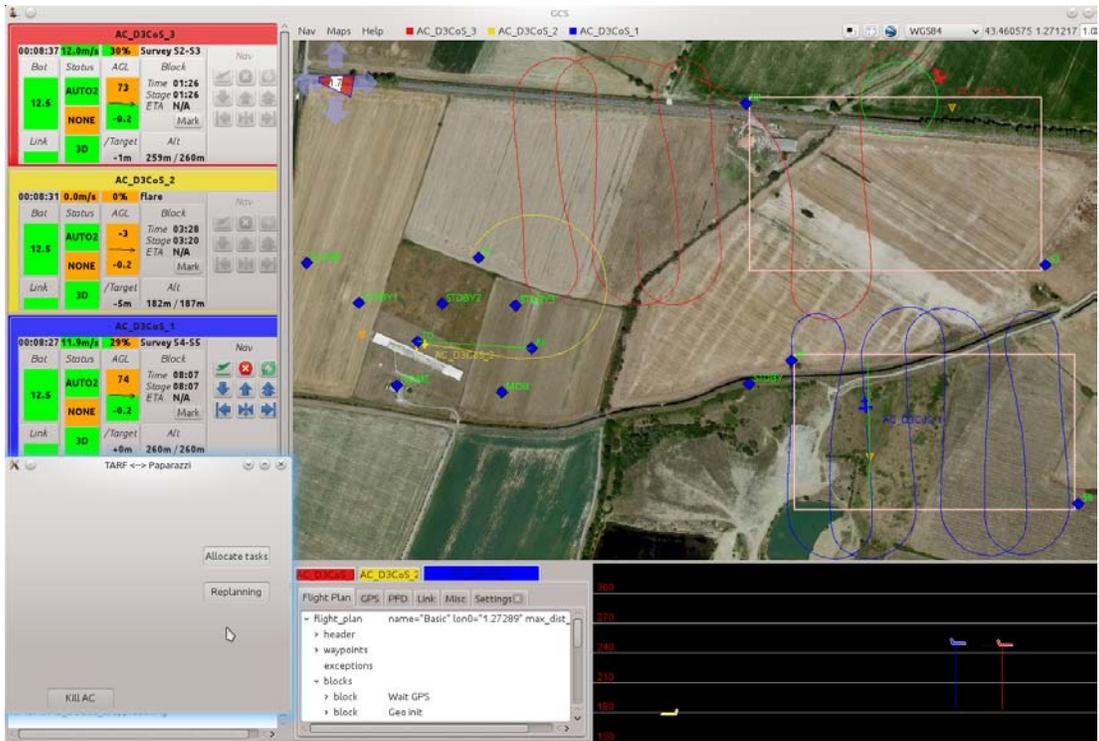


Figure 14. Updating TARF: the red aircraft continues the S2-S3 section surveillance task of the yellow one

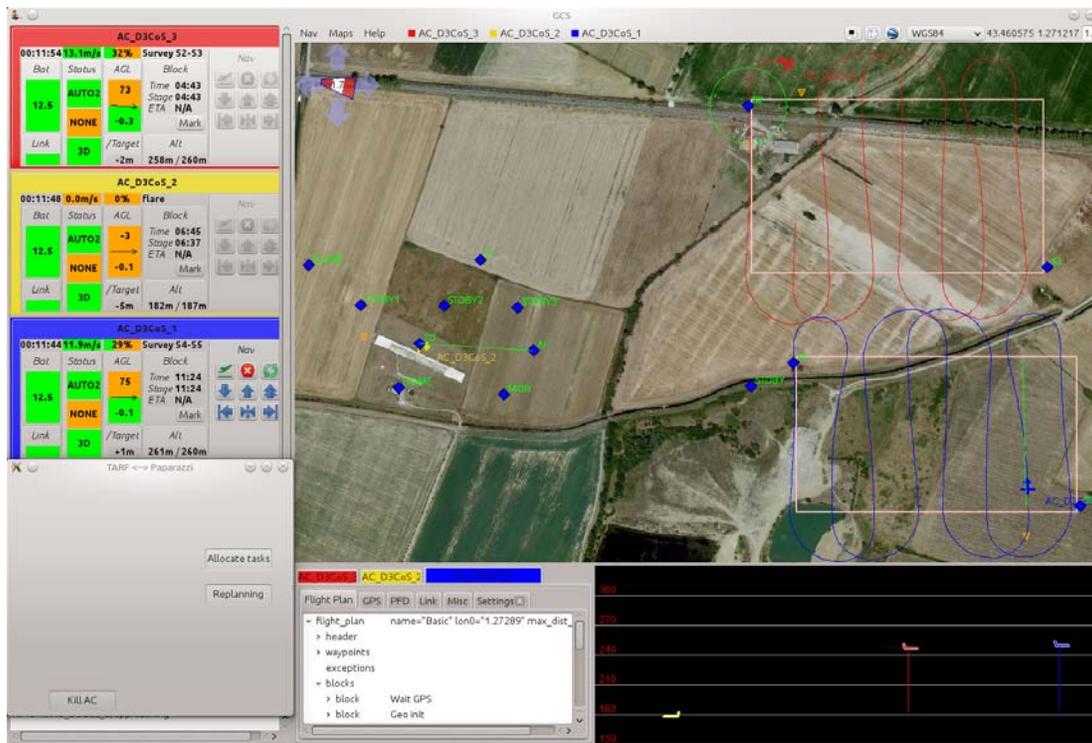


Figure 15. Two aircrafts have almost finished their tasks and they are ready to land

## References

- <sup>1</sup>Hausamann, D., Zirng, W., Schreier, G., and Strobl, P., "Monitoring of gas pipelines—a civil UAV application," *Aircraft Engineering and Aerospace Technology*, Vol. 77, No. 5, 2005, pp. 352–360.
- <sup>2</sup>Karlsson, R., Schon, T., Tornqvist, D., Conte, G., and Gustafsson, F., "Utilizing model structure for efficient simultaneous localization and mapping for a UAV application," *Aerospace Conference, 2008 IEEE*, IEEE, 2008, pp. 1–10.
- <sup>3</sup>Mayer, S., Hattenberger, G., Brisset, P., Jonassen, M. O., and Reuder, J., "A no-flow-sensor wind estimation algorithm for Unmanned Aerial Systems," *International Journal of Micro Air Vehicles*, Vol. 4, No. 1, 2012, pp. 15–30.
- <sup>4</sup>Hardin, P. J. and Jackson, M. W., "An unmanned aerial vehicle for rangeland photography," *Rangeland Ecology & Management*, Vol. 58, No. 4, 2005, pp. 439–442.
- <sup>5</sup>Beard, R. W., Kingston, D., Quigley, M., Snyder, D., Christiansen, R., Johnson, W., McLain, T., and Goodrich, M., "Autonomous vehicle technologies for small fixed-wing UAVs," *Journal of Aerospace Computing, Information, and Communication*, Vol. 2, No. 1, 2005, pp. 92–108.
- <sup>6</sup>Tu, H. and Du, X., "The Design of Small UAV Autopilot Hardware System Based on DSP," *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, Vol. 3, IEEE, 2010, pp. 780–783.
- <sup>7</sup>Lim, H., Park, J., Lee, D., and Kim, H. J., "Build your own quadrotor: Open-source projects on unmanned aerial vehicles," *Robotics & Automation Magazine, IEEE*, Vol. 19, No. 3, 2012, pp. 33–45.
- <sup>8</sup>Brisset, P., Drouin, A., Gorraz, M., Huard, P., and Tyler, J., "The paparazzi solution," *MAV2006, Sandestin, Florida*, 2006.
- <sup>9</sup><http://paparazzi.enac.fr>, "PAPARAZZI, Open-source hardware and software project," .
- <sup>10</sup>Hu, W. and Liu, G., "Dynamic group method based on health management applied to mission tasking for multi UAVs," *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, 2012, pp. 4118–4121.
- <sup>11</sup>Palmer, D., Kirschenbaum, M., Murton, J., Zajac, K., Kovacina, M., and Vaidyanathan, R., "Decentralized cooperative auction for multiple agent task allocation using synchronized random number generators," *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, Vol. 2, IEEE, 2003, pp. 1963–1968.
- <sup>12</sup>Gancet, J., Hattenberger, G., Alami, R., and Lacroix, S., "Task planning and control for a multi-UAV system: architecture and algorithms," *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, IEEE, 2005, pp. 1017–1022.
- <sup>13</sup>Bellingham, J., *Multi-task allocation and path planning for cooperating UAVs*, Ph.D. thesis, Citeseer, 2001.
- <sup>14</sup>Chanthery, E., Barbier, M., Farges, J., and Center-FRANCE, O., "Planning algorithms for autonomous aerial vehicle," *16th IFAC World Congress*, Vol. 16, 2005.
- <sup>15</sup>Gerkey, B. and Mataric, M., "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, Vol. 23, No. 9, 2004, pp. 939–954.

<sup>16</sup>Pongpunwattana, A. and Rysdyk, R., “Real-time planning for multiple autonomous vehicles in dynamic uncertain environments,” *Journal of Aerospace Computing, Information, and Communication*, Vol. 1, No. 12, 2004, pp. 580–604.

<sup>17</sup>Bertuccelli, L. F., Choi, H.-L., Cho, P., and How, J. P., “Real-time multi-UAV task assignment in dynamic and uncertain environments,” *AIAA Guidance, Navigation, and Control Conference, Chicago, Illinois*, 2009.

<sup>18</sup>Grotli, E. I. and Johansen, T. A., “Task assignment for cooperating UAVs under radio propagation path loss constraints,” *American Control Conference (ACC), 2012*, IEEE, 2012, pp. 3278–3283.

<sup>19</sup>Vokrinek, J., Komenda, A., and Pechoucek, M., “Abstract architecture for task-oriented multi-agent problem solving,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, Vol. 41, No. 1, 2011, pp. 31–40.

<sup>20</sup>Smith, R. G., “The contract net protocol: High-level communication and control in a distributed problem solver,” *Computers, IEEE Transactions on*, Vol. 100, No. 12, 1980, pp. 1104–1113.

<sup>21</sup><http://www.tls.cena.fr/products/ivy>, “IVY, A simple protocol and a set of open-source (LGPL) libraries,” .

<sup>22</sup><http://www.d3cos.eu/index.php/component/content/frontpage>, “The international project D3CoS,” .