Ground stations for systems of multiple UAVs interacting with sensors and actuators

A. Ollero

Ground stations for systems of multiple UAVs interacting with sensors and actuators in the environment

Anibal Ollero [(1) (2)] and Iván Maza [(1)]

aollero@cartuja.us.es

aollero@catec.aero

(1) Robotics, Vision and Control Group (GRVC), Universidad de Sevilla

(2) Center for Advanced Aerospace Technologies FADA-CATEC

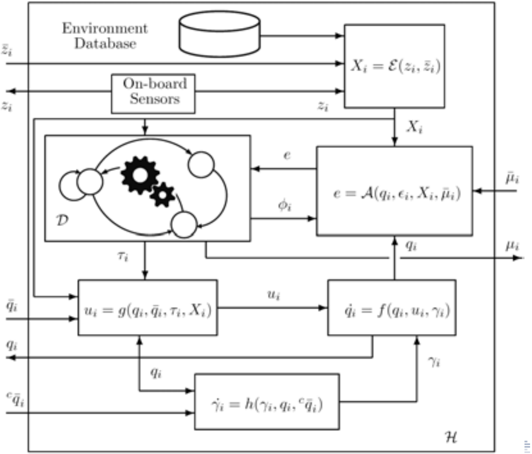
Outline

Objectives

- Devise a distributed architecture for the autonomous cooperation of multiple UAVs with the following execution level autonomous capabilities:
  - Take-off and landing.
  - Hovering.
  - Go to a given location (and activate the on-board instrument if required).
- **Design a human machine interface application for the platform.**
- Develop a software implementation of the architecture and the **human machine interface**.
- Test the implementation with a real multi-UAV platform in different types of missions.

Classification of Multi-UAV Systems

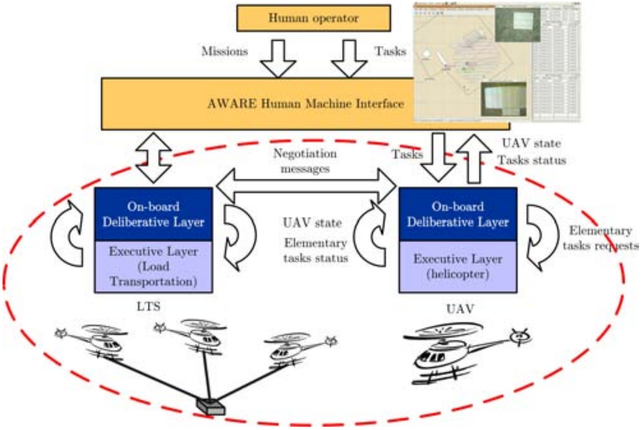
- Physical coupling
- Formations
- Swarms
- Intentional cooperation



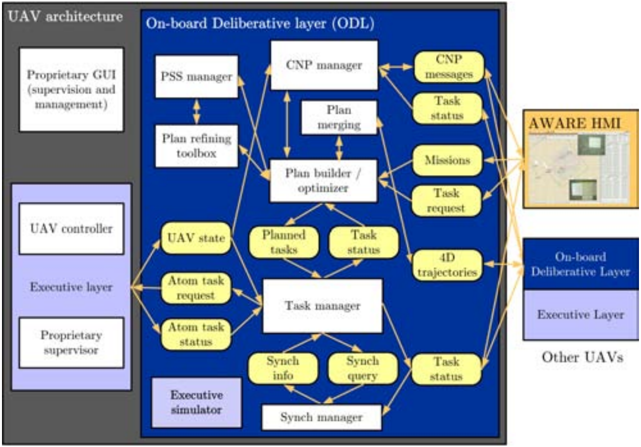
Introduction

- Centralized / Decentralized Decision trade-off:
  - Computational power and scalability.
  - Knowledge's scope and accessibility.
- Objective: distributed approach.
  - Why?
    - Scalability.
    - Robustness.
- Issues:
  - Cooperative decision making.
  - Knowledge representation and information fusion.

Global Architecture



ODL Internal Architecture



Task Model (ODL)

$\tau_i^k = (\lambda, {}^{-}\Omega, \Omega^{+}, \varepsilon, \Pi)$

Task with unique identifier k allocated to the i-th UAV, where

- λ: type of task.
- ${}^{-}\Omega$: set of preconditions of the task.
- Ω^{+}: set of postconditions of the task.
- ε: status evolution of the task
- $\Pi = \{\pi^1, \pi^2, \ldots, \pi^m\}$: set of m parameters which characterizes the task.

Task Model (ODL)

$$\tau_i^k = (\lambda, ^- \Omega, \Omega^+, \varepsilon, \Pi)$$

Status (ε)	Description
EMPTY	No task
SCHEDULED	The task is waiting to be executed
RUNNING	The task is in execution
CHECKING	The task is being checked against inconsistencies and static obstacles
MERGING	The task is in the plan merging process to avoid conflicts with the trajectories of other UAVs
ABORTING	The task is in process to be aborted. If it is finally aborted, the status will change to ABORTED, and otherwise will return to RUNNING
ABORTED	The task has been aborted (the human operator has aborted it or the UAV was not able to accomplish the task properly)
ENDED	The task has been accomplished properly

Task Model (ODL)

$$\tau_i^k = (\lambda, {}^-\Omega, \Omega^+, \varepsilon, \Pi)$$

Task type(λ):

`TAKE-OFF, LAND, GOTO, GOTOLIST, DEPLOY,`
`TAKE-SHOT, WAIT, SURV, DETECT, TRACK, HOME`

Example

Parameters of a surveillance task

Parameters (Π)	Description
π^1 (Polygon)	The set of vertices defining the polygon of the area to be covered by the UAV
π^2(Altitude)	Altitude (m) for the flight (ellipsoid-based datum WGS84)
π^3(Speed)	Specified speed (m/s) for the flight
π^4(Overlapping)	Desired overlapping in percentage between consecutive rows of the zigzag pattern used to cover the area

ODL Internal Architecture

- Task manager
  - Manages the interface between ODL and EL.
  - Supports dynamic task insertion and abortion mechanisms.
- Synchronization manager.
  - Deals with preconditions $({}^{-}\Omega)$ and postconditions (Ω^{+}) among different UAVs.
- Perception subsystem.
  - Fully distributed probabilistic framework adopted to provide estimations about the objects in the environment.

ODL Internal Architecture

- Plan builder / optimizer.
  - Offline planning based on the EUROPA framework developed at NASA's Ames Research Center.
  - Online planning: given a set of n_m motion tasks $\{\tau_i^k / k = 1 \ldots n_m\}$ computes the order that minimizes the execution cost $C_i = \sum_{k=1}^{n_m - 1} c_i^{k,k+1}$, where $c_i^{k,k+1}$ is the motion cost between the locations associated to the tasks τ_i^k and τ_i^{k+1}.
- Executive simulator.
  - Used for debugging purposes.

Location Monitoring

- Associated task type: $\lambda = \texttt{TAKE-SHOT}$.
- Goal: Monitor a given location $[\ x_0 \quad y_0 \quad z_0\ ]^T$.
- Fixed and known orientation of the on-board camera.
- Assuming a given altitude $z = z_\Pi$ and heading for the observation, the coordinates of the UAV will be given by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 + (z_\Pi - z_0)v_x/v_z \\ y_0 + (z_\Pi - z_0)v_y/v_z \\ z_\Pi \end{bmatrix}$$

with

$$\mathbf{v}_G = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \frac{1}{s} \underbrace{\mathbf{R}_G^U \mathbf{R}_U^C}_{\text{rotation matrices}} \underbrace{\begin{bmatrix} \frac{v - v_0}{\alpha_v} \\ \frac{u}{\alpha_u} - \frac{u_0}{\alpha_v} - \gamma \frac{v - v_0}{\alpha_u \alpha_v} \\ 1 \end{bmatrix}}_{\text{camera parameters}}.$$

Location Monitoring

Camera Parameters

- (u_0, v_0): coordinates of the principal point.
- α_u and α_v: scale factors in image u and v axes.
- γ: skewness of the image axes.
- If the location should be in the center of the field of view (default), then

$$\mathbf{m} = [\begin{array}{cc} u & v \end{array}]^T = [\begin{array}{cc} w/2 & h/2 \end{array}]^T$$

for a camera resolution of $w \times h$ pixels.

Rotation matrices

- $\mathbf{R}_U^C$: from camera to UAV frame.
- $\mathbf{R}_G^U$: from UAV to global frame.

Object Monitoring

- Associated task type: $\lambda = \texttt{TRACK}$.
- Goal: Monitor an object k with associated covariance matrix $\mathbf{C}_k$ and estimated state $\mathbf{x}_k(t) = \begin{bmatrix} \mathbf{p}_k(t) & \dot{\mathbf{p}}_k(t) & \theta_k \end{bmatrix}^T$.
- According to (Forssén, 2004)

$$\mathbf{M}_k = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} = \frac{1}{4}\mathbf{C}_k^{-1},$$

where $\mathbf{M}_k$ is the matrix describing the shape of an ellipse $(\mathbf{x}-\mathbf{k})^T\mathbf{M}(\mathbf{x}-\mathbf{k}) = 1$.

- Major axis of the ellipse:

$$\mathbf{v}_1 = \begin{cases} \frac{1}{\sqrt{(\lambda_1 - m_{22})^2 + m_{12}^2}} \begin{bmatrix} \lambda_1 - m_{22} \\ m_{12} \end{bmatrix} & \text{if} \quad m_{11} \geq m_{22} \\ \frac{1}{\sqrt{(\lambda_1 - m_{11})^2 + m_{12}^2}} \begin{bmatrix} m_{12} \\ \lambda_1 - m_{11} \end{bmatrix} & \text{otherwise} \end{cases}$$

with $\lambda_{1,2} = \frac{(m_{11} + m_{22}) \pm \sqrt{(m_{11} - m_{22})^2 + 4m_{12}^2}}{2}$.

Object Monitoring

Example

Waypoint computation for object monitoring tasks

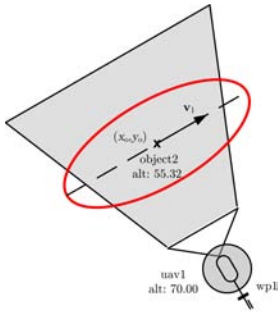


Figure: Object monitoring with a single UAV

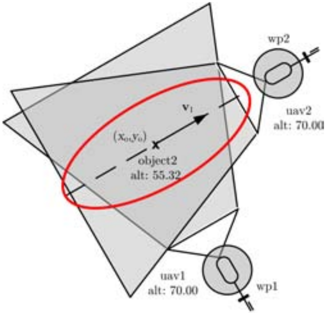


Figure: Locations for object monitoring with two UAVs

Object Deployment

- Associated task type: $\lambda =$ DEPLOY.
- Task decomposition:
  - Go to the waypoint.
  - Go down until an altitude of h_d meters with respect to the ground is reached.
  - Activate the on-board device for dropping the object.
  - Go up to the initial waypoint altitude.
- Given a deployment task τ_i^k allocated to the i-th UAV:
  - $\tau_i^k \rightarrow \{{}^1\hat{\tau}_i^k, {}^2\hat{\tau}_i^k, {}^3\hat{\tau}_i^k, {}^4\hat{\tau}_i^k\}$ ($\hat{\lambda}^k =$ GOTO)

Surveillance Missions

- Associated task type: $\lambda = \mathtt{SURV}$.
- Goal: To cover a convex area with one or several UAVs.
- Problems to solve:
  1. Area partition:
     - Input: relative capabilities (distributed computation) and locations of the UAVs.
     - Output: sub-areas allocated to the UAVs.
  2. Pattern to cover each sub-area.
     - Output: waypoint list for each UAV.
     - Criteria: minimize the number of required turns.

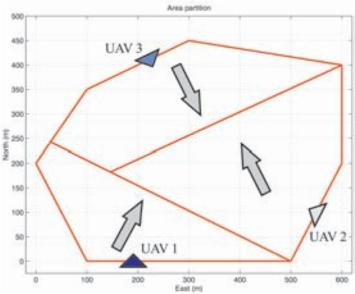
Surveillance Missions

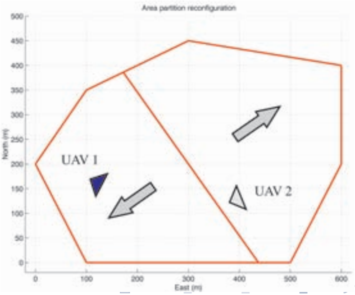
- Associated task type: $\lambda = \texttt{SURV}$.
- Goal: To cover a convex area with one or several UAVs.
- Problems to solve:
  1. Area partition:
     - Input: relative capabilities (distributed computation) and locations of the UAVs.
     - Output: sub-areas allocated to the UAVs.
  2. Pattern to cover each sub-area.
     - Output: waypoint list for each UAV.
     - Criteria: minimize the number of required turns.

Surveillance: Area Partition Example

Table: Initial coordinates, sensing width and relative capabilities of the UAVs.

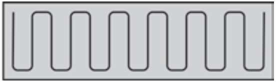
	x_G(m)	y_G(m)	z_G(m)	w_i(m)	c_i(%)
UAV1	190.00	0.00	29.00	24.02	24.92
UAV2	550.00	100.00	34.00	25.45	41.81
UAV3	225.38	412.69	20.00	20.00	33.27



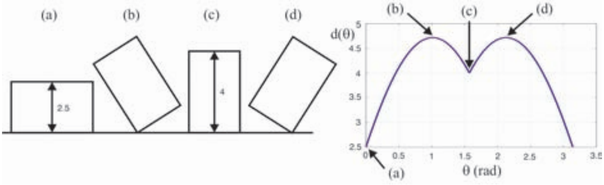


Surveillance: Coverage Pattern

- Zigzag faster than spiral pattern in general (Huang, 2001).
- Find the sweep direction that minimizes the number of turns:



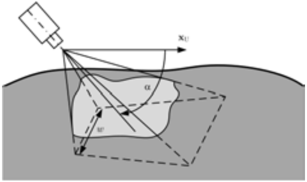
- Number of turns proportional to the diameter function of the polygon



- Diameter function minimum ⇒ test only the sweep directions perpendicular to an edge of the perimeter.

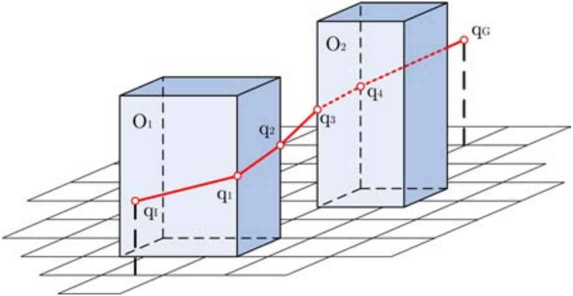
Surveillance: Coverage Pattern

- Distance between rows:
  - Sensing width (w).
  - Specified overlapping (%) between consecutive rows.



Static Obstacles Avoidance

- Initial elevation map available.
- Goal: find shortest paths free of obstacles from configuration q_I to q_G.
- Obstacles are modelled as boxes on a plain terrain $\Rightarrow$ combinatorial algorithms (exact and complete) can be applied:
  - Shortest-path roadmap in three dimensions (Jiang et al., 1993).
- Output: a waypoint list avoiding the obstacles.



Overview

Problem Statement

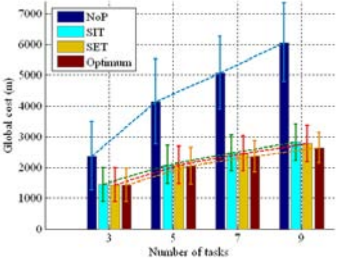
Given a set of tasks, decide in a distributed manner which UAV will execute each task

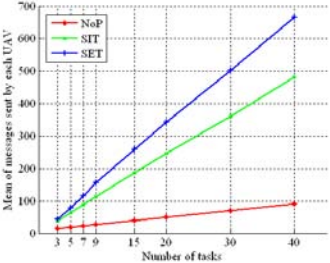
- Three market-based approach algorithms designed and simulated:
  - SIT and SET: independent tasks.
  - S+T: tasks requiring explicit cooperation (communication relay).
- Algorithms based on the *Contract Net Protocol*.
- An auction mechanism is used to allocate the tasks:
  - The auctioneer announces the tasks to be executed.
  - Each UAV bids for the tasks.
  - Each task is allocated to the UAV with the “best” bid.
  - Partial local plans are used to compute the bid (insertion cost of the new task in the current plan).

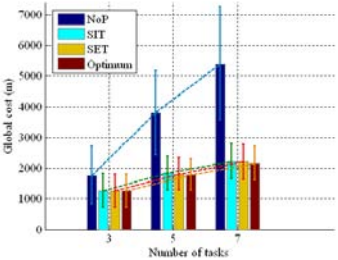
SIT and SET Algorithms

- SIT algorithm:
  - Individual tasks negotiated.
- SET algorithm:
  - Each UAV can generate group of tasks for the negotiation.
  - If all the groups have one task $\Rightarrow \mathrm{SET} \equiv \mathrm{SIT}$.
- Simulation enviroment:
  - Code of the algorithms in the simulations: used later with the real platform.
  - Scenario: world of 1000×1000 meters.
  - Mission: visit the waypoints and return home.
  - Waypoints located at random and 100 runs per configuration to generate the statistical performance metrics.
  - Global cost: sum of the individual costs of the UAVs to achieve the mission.
  - Results compared with *NoP* algorithm (no local plan used when bidding) and the minimum cost.

SIT and SET Simulation Results







SIT vs SET

SIT algorithm:

- Better trade-off between solution computed and messages required.
- Chosen for the experimental validation.

Approach Adopted

- Main criteria: guarantee maximum safety.
- Assumption: hovering or pseudo-hovering (circles) capabilities.
- Considering the motion of the i-th UAV:
  - State s_i^1: stationary flight at waypoint P.
  - State s_i^2: flying between waypoints P^k and P^{k+1} (path Δ_i^k).
- Conflicts with the j-th UAV ($s_i^1 \rightarrow s_i^2$ transition):
  - Type A (if $s_j = s_j^1$): between next path Δ_i^k and current position of the j-th UAV.
  - Type B (if $s_j = s_j^2$): between next path Δ_i^k and path Δ_j^l being currently followed by the j-th UAV.

Problem formulation

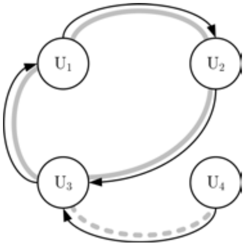
Avoid conflicts of type A and B in the transitions $s_i^1 \rightarrow s_i^2 \;\forall\; i = 1, \ldots, n$

Distributed Method

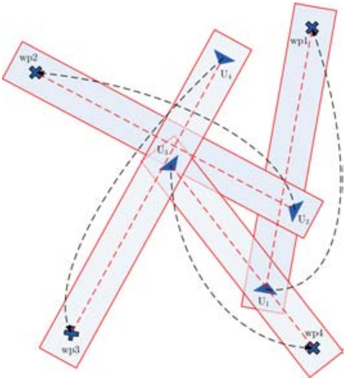
- Task state $\epsilon^k = \texttt{MERGING}$ inserted between SCHEDULED and RUNNING.
- Distributed algorithm based on *request* and *reply* messages. Transitions:
  - χ_1: $\epsilon^k = \texttt{MERGING}$
    - Broadcast a *request* message.
  - χ_2: $receive(m)_{j,i}$ with $m = request(x, \Delta_j^l)$
    - If no conflicts ⇒ answer with *reply* message.
  - χ_3: $receive(m)_{j,i}$ with $m = reply$
    - If *reply* messages received from all the UAVs ⇒ proceed with the execution.
- Lemma: The method guarantees that a path Δ_i^k is executed by the i-th UAV only if it is clear of other UAVs (proof provided).

Deadlocks Solution

- Distributed deadlock detection algorithm:
  - Based on probe messages (Lee and Kim, 2001).
  - Assumption: messages received correctly in order.
  - Performance:
    - All true deadlocks detected.
    - Deadlocks not reported falsely.
- Example:

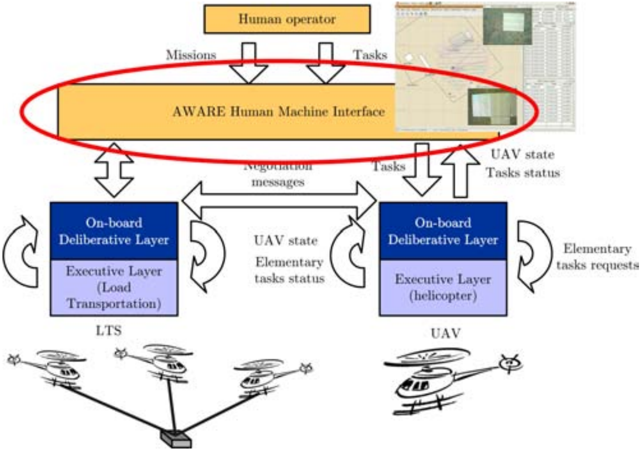


Associated wait-for graph



Global Architecture

Human Machine Interface Application



Introduction

- Graphical application to:
  - Monitor the platform state: UAVs, sensor nodes, cameras, etc.
  - Insert tasks and missions.
- Operator's workload significantly decreased during the missions due to the autonomy of the platform:
  - Inter-UAV collision avoidance.
  - Task allocation process.
  - Static obstacles avoidance.
  - Plan refining tools.
- Research done in this area:
  - Study the improvement associated to the use of multimodal interfaces for the alerts awareness of the user.

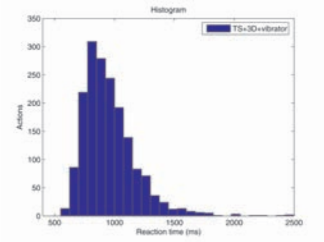
Multimodal Setup

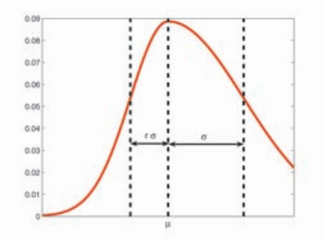
- Three different modalities considered:
  - Aural: speakers (speech synthesis) and headset (3D audio).
  - Visual: three touchscreens.
  - Tactile: three wireless vibrators (chest and left and right arms).

- Tests performed:
  - "Yes" and "No" buttons appear in random positions on the several touchscreens during a programmed period.
  - Operator's task: Press only "Yes" buttons as soon as possible (i.e. as an alert acknowledge).
  - Reaction time and errors were measured.

Analysis of the Tests

- Histograms computed for each setup.
- To capture the temporal asymmetric distributions, the Asymmetric Gaussians (AG) have been considered.

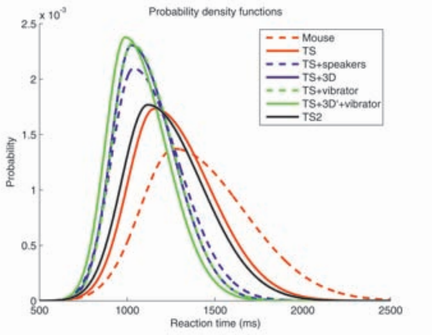




$$\varphi_{\mu,\sigma^2,r}(x) = \frac{2}{\sigma(r+1)\sqrt{2\pi}} \begin{cases} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) & \text{if } x > \mu, \\ \exp\left(-\frac{(x-\mu)^2}{2r^2\sigma^2}\right) & \text{otherwise} \end{cases}$$

Summary of the Results

- Experimental setups:
  - Mouse.
  - TS: touchscreens.
  - TS+speakers: touchscreens and conventional sound.
  - TS+3D: touchscreens and 3D sound.
  - TS+vibrator: touchscreens and tactile.
  - TS+3D+vibrator: touchscreens, 3D sound and tactile.
  - TS2: touchscreens test repeated.



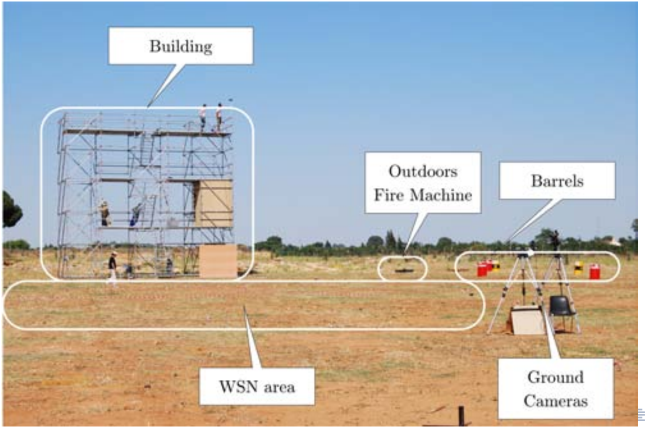
- The operator's performance is improved adding modalities.
- Different relevance of the modalities.

Virtual head mounted display

Head-tracking for 3D environments

Experimentation Scenario

- AWARE Project framework.
- Location: Iturri Group facilities in Utrera (Seville).
- Validation scenarios: disaster management and filming.



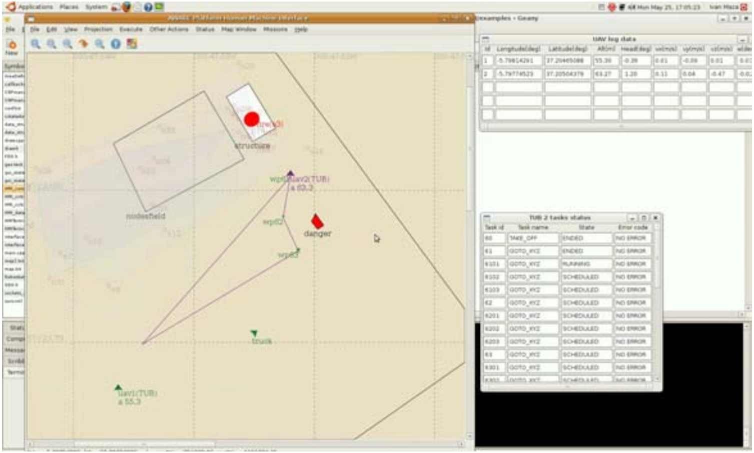
AWARE Platform Components

Scheduling and Metrics

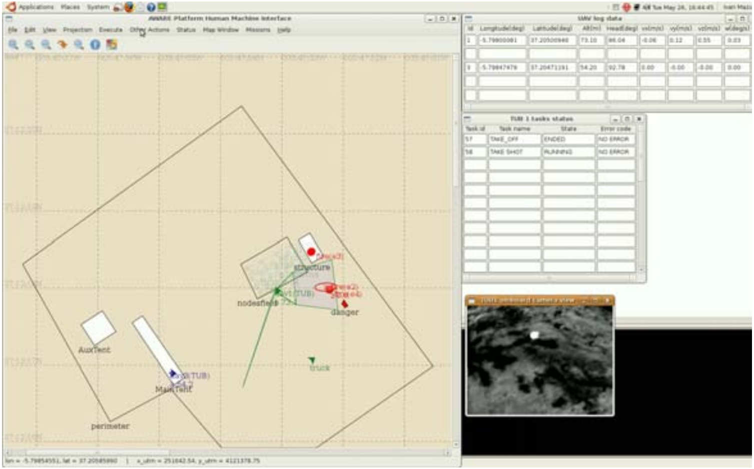
Mission	Date	Brief description	HMI	UAV	GCN	WSN	FT
1	16th April 08	Node deployment	✓	✓		✓	
2	25th May 09	Firemen tracking	✓	✓	✓	✓	
3	25th May 09	Firemen tracking	✓	✓	✓	✓	
4	25th May 09	Surveillance	✓	✓			
5	25th May 09	Node deployment & fire monitoring	✓	✓		✓	✓
6	25th May 09	Node deployment & fire monitoring	✓	✓		✓	✓
7	26th May 09	Fire monitoring	✓	✓		✓	✓
8	26th May 09	Surveillance	✓	✓			✓
9	28th May 09	Load transportation	✓	✓			
10	28th May 09	Node deployment	✓	✓		✓	
11	28th May 09	Fire monitoring	✓	✓	✓	✓	✓
12	28th May 09	Surveillance	✓	✓			✓

Mission #	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**Total**
Tasks received from the HMI	9	10	8	12	12	6	8	3	4	6	8	86
Elementary tasks sent to the executive	9	10	22	21	21	6	28	3	7	6	18	151
Tasks generated by the plan refiner	4	4	18	13	13	3	24	0	3	3	14	99
Coordination messages interchanged	12	12	66	34	34	0	78	0	0	0	58	294
Potential conflicts managed successfully	6	6	18	17	17	0	24	0	0	0	14	102

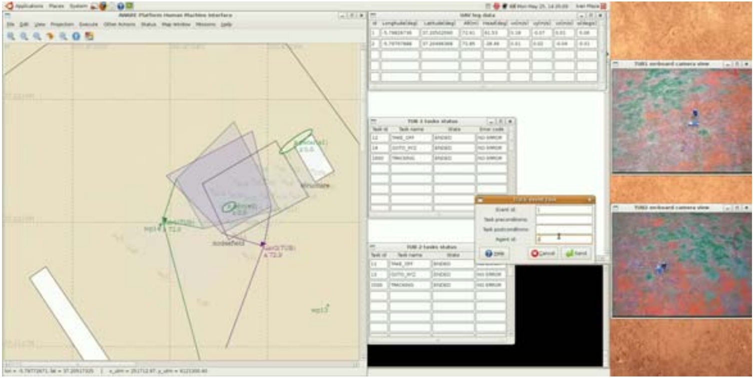
Sensor Deployment Mission



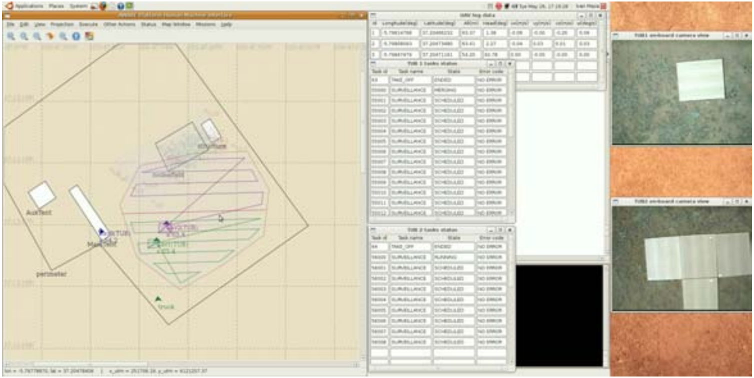
Fire Confirmation and Extinguishing



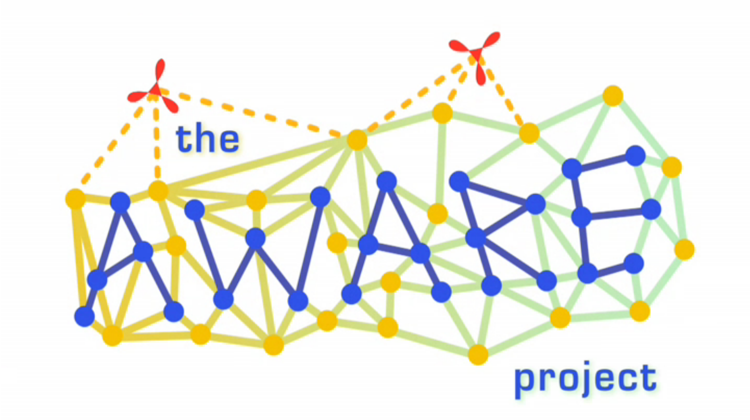
People Tracking Mission



Surveillance Mission



AWARE Project Video



Conclusions

- Field validation of a very complex system in a context of specific end-user requirements.
- Research contributions in the following areas:
  - Autonomous complex task decomposition.
  - Distributed task allocation algorithms.
  - Distributed plan merging to avoid inter-UAV conflicts.
  - Human machine multi-modal interfaces.
- Strong emphasis on the experimental validation of the architecture. Lessons learned:
  - Time synchronization (NTP), data marshalling, common coordinate frames, etc.
  - Relevance of the integration meetings to debug the software interfaces using simulators (more than 20 in the last year).
  - Successful mission execution requires reliable:
    - Decisional level software.
    - UAV platform (TUB and FC in AWARE).

Future Developments

- Test the distributed architecture with other UAVs (including fixed wing).
- Integrate the EUROPA planner for online planning (documentation still poor).
- Derive theoretical performance metrics of the distributed plan merging algorithm (number of deadlocks, deadlock duration, etc.).
- Integrate head tracking technologies in the human machine interface:
  - HMI aware of the operator's state.
  - 3D views control.
- Experimental validation of the S+T algorithm for task allocation under communication range constraints.
- Extend the plan refining tools to cover a wider spectrum of missions.

CATEC UAVs

FADA FUNDACIÓN ANDALUZA PARA EL DESARROLLO AEROESPACIAL

CATEC CENTRO AVANZADO de TECNOLOGÍAS AEROESPACIALES

Multi-vehicle systems lab

- Multi-vehicle testbed
  - “Rapid prototyping of multi-vehicle systems”
  - 14x14x5 meters
  - Based on VICON cameras
    - More than 10 mobile objects/humans at the same time
    - 3D position and pose accurate and high rate information
  - Up to 10 flying vehicles at the same time
    - 8 Hummingbird quadrotors: 200gr payload (small visual camera)
    - 2 Pelican quadrotors: 500gr payload (Hokuyo laser with onboard computer)
  - Suitable for testing: Distributed planning, task allocation, Share of common resources
  - ATM: High level mission decomposition, Task-based guidance, Ground and aerial cooperation