2023 IEEE 26th International Conference on
Intelligent Transportation Systems (ITSC)
24-28 September 2023. Bilbao, Bizkaia, Spain

Cooperative architecture using air and ground vehicles for the search and recognition of targets

M. Theunissen, H. Pousseur, P. Castillo, A. Correa Victorino

2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3503-9946-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITSC57777.2023.10421916

***Abstract*— A cooperative navigation architecture for the search and recognition of targets using aerial and ground vehicles is proposed in this paper. Two main tasks are conceived; an aerial monitoring of a surface for searching targets, and a target ground recognition. First, the aerial drone tracks autonomously a trajectory, computed to cover all the surface to monitor, to search for targets using vision algorithms. Once detected, its relative position is sent to the cooperative architecture. After the aerial drone has covered the entire area, the architecture computes and assigns to each ground vehicle the closest target found. Then, each ground vehicle navigates autonomously avoiding obstacles to its assigned target. To verify the success of the mission, the aerial vehicle flies over the dynamic center of mass of the ground vehicles. Real-time experiments are carried out to validate the proposed architecture. Main results, depicted in some graphs, corroborate the good performance in closed loop.**

I. Introduction

In recent years, works on cooperation between heterogeneous multi-robot systems have multiplied. By sharing their capacities, such systems aim to gain in flexibility, efficiency and robustness and tend to achieve total autonomy in dynamic environments. Some frameworks for cooperative air and ground vehicles have been mostly used for collaborative path-planning and obstacle avoidance [1], target tracking [2], exploration and mapping missions [3].

A cooperative robotic system for target searching and recognition in an unknown environment is an open challenge that requires a close cooperation between robots. Some works addressing this problem use a preliminary exploration step to map the environment. In [4], the authors proposed to take advantage of the UAV (unmanned aerial vehicle) aerial view to detect obstacles that are mapped. A path planning algorithm then determines a clear path that will be follow by a UGV (unmanned ground vehicle) to reach a target point. In this work, the navigation of the ground robot depends on the created map. It is also assumed that the UAV can see the environment entirely which is difficult to do in real scenarios.

Other works, such as [3] or [5], suggest a cooperative environment exploration. In [3], the authors proposed a team of UAV and UGV with LiDARs performing active exploration to build a 3D map of an unknown environment. However, the proposed scenario is performed sequentially thus weakening the cooperation. [5] improved this concept by proposing a cooperative framework to gain in efficiency and robustness, reducing drastically the robots' movements. The environmental scan merges sensors from different natures: a stereo camera and a LiDAR that benefit the mapping.

In [6], the authors proposed to make ground robots navigation in a GPS denied environment thanks to a semantic map built by a UAV. This work highlights the importance of the UGVs' local planner to avoid obstacles in real experiments. This point raises the question of the necessary accuracy of the map built during the exploration and the long time allocated to this step. In [7], the authors have developed a cooperative navigation between one UGV and one UAV for targets search in an unknown environment with obstacles. The UAV first scans the environment with a constant altitude to find all targets that are sent to the ground robot. The UGV then reaches the targets on a first-come-first-served basis thanks to an online trajectory generator to avoid obstacles. This work has the advantages of reducing communication between robots. However, the proposed scenario is not optimized in terms of navigation time and is not compatible with multiple ground robots to speed up the mission.

This work focuses on air-ground cooperative systems for targets searching and recognition. The main interest comes from the great complementarity between the capacities (payload capacity, speed, stability, detection, communication, etc.) of UAVs and UGVs which make them powerful to complete complex tasks. Nevertheless, perception and control issues are still open for both systems (aerial and ground) to have a good performance and guarantee the success of the mission. In addition, in this work, we propose a cooperative architecture to ensure the coherence of the robots' activities to optimize the mission. Our architecture involves a UAV drone and several independent UGV. In addition, this platform has been designed to make common navigation possible in an unmapped environment. In our work, we make sure to cover, in an aerial mode, all the surface to be monitored in order to find all the unknown targets. Moreover, the found targets location is sent to the cooperative architecture to assign tasks to each ground vehicle to reach the closest target. Our cooperative architecture is validated in a physical twin in real-time experiments.

The paper is organized as follows: the problem statement and main dynamics equations for aerial and ground vehicles are presented in section II. In section III, the cooperative navigation architecture is explained in details. The experimental setup and the obtained results are shown in section

This work has been partially supported by ROBOTEX 2.0 (Grants ROBOTEX ANR-10-EQPX-44-01 and TIRREX ANR-21-ESRE-0015) - France,

M. Theunissen, H. Pousseur, A. Correa Victorino, P. Castillo are with Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems) CS 60319 - 60203 Compiègne Cedex. (mathilde.theunissen, hugo.pousseur, castillo, acorreav)@utc.fr

IV. Finally, conclusions and some future works are given in section V.

II. Problem Statement and Preliminaries

The success of some civil missions hinges on the collaboration between aerial and ground vehicles. For instance, the accomplishment of targets finding missions (like human lost in the forest) can be harduous and time expensive for homogeneous robots. Thus, the challenge to use an heterogeneous robots cooperation scheme is to avoid targets' occlusions (produced by obstacles) and develop the mission in a short time. Some works in literature propose solution for this kind of civil applications using only one kind of robots, others include both configurations but some solutions are only validated numerically.

Our solution proposes to use heterogeneous robots (aerial and ground) to perform monitoring tasks. The validation scenario is presented in Figure 1, where an aerial vehicle must cover an area to find specific targets (whose position is unknown). Then, the architecture computes closest target position from each ground vehicle presented in the scenario awarding the task of going to recognize the objective. The monitoring, searching and recognition tasks are done by the robots in autonomous mode. In this work, we assume that all robots know their position in a shared global reference frame and that all robots are connected to the same functional network in order to communicate.

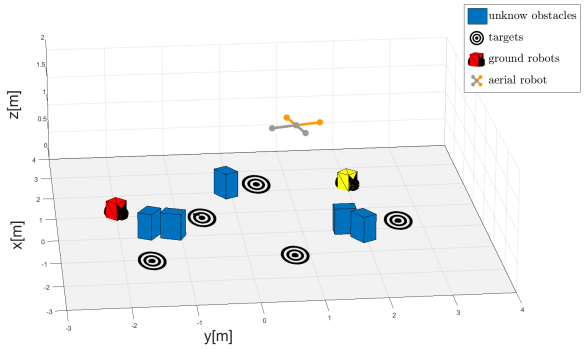

Fig. 1: Search and recognition of targets using heterogeneous vehicles.

In this work, three robots are used; one UAV with a four rotors configuration and two homogeneous unicycle-type UGV as shown in figure 2. The dynamic or kinematic models of each vehicle will be introduced in the following.

A. Aerial dynamic model

For the aerial mission, a quadcopter vehicle is considered. This vehicle can be represented as a rigid body evolving in three dimensions thanks to external actions: a main thrust u_q and three torques $\boldsymbol{\tau}_q = [\tau_\psi, \tau_\theta, \tau_\phi]$ [8]. Its mathematical equations can written using the quaternion representation as

$$m_q \ddot{\boldsymbol{\xi}}_q = \boldsymbol{q} \otimes \boldsymbol{F}_{th} \otimes \boldsymbol{q}^* + m_q \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \boldsymbol{\zeta}_p \quad (1)$$

$$J \dot{\boldsymbol{\Omega}} = \boldsymbol{\tau}_q - \boldsymbol{\Omega} \times J\,\boldsymbol{\Omega} + \boldsymbol{\zeta}_\Omega \quad (2)$$

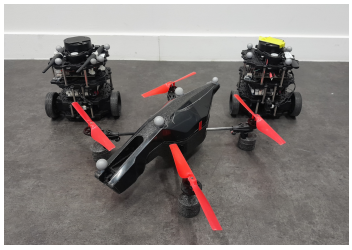

Fig. 2: Ground and aerial robots in cooperation.

where

$$\boldsymbol{F}_{th} \coloneqq \begin{bmatrix} 0 \\ 0 \\ u_q \end{bmatrix} \quad (3)$$

is the vehicle's thrust force vector in the body frame; its magnitude is often considered as a control input (u_q). However, note its effect in the inertial frame directly depends on the system's attitude. $\boldsymbol{\xi}_q = [x_q, y_q, z_q]^T$ denotes the aerial vehicle position in the inertial frame $\mathcal{I}$, m_q the mass of the aerial vehicle and g the gravity force. The term $\boldsymbol{q} \in \mathbb{H}$ is the system's orientation in unit quaternion form and $\boldsymbol{q}^*$ denotes its conjugate term. J is the inertia matrix, $\boldsymbol{\Omega}$ denotes the angular velocity and the terms $\boldsymbol{\zeta}_p$ and $\boldsymbol{\zeta}_\Omega$ are the external disturbances.

B. Ground kinematic model

Ground unicycle-type robots are used for the mission. Each ground vehicle i is represented using the following kinematic equations and assuming that their altitude z_{g_i} is not changing.

$$\begin{bmatrix} \dot{\boldsymbol{\xi}}_{g_i} \\ \dot{\psi}_{g_i} \end{bmatrix} = \begin{bmatrix} \cos\psi_{g_i} & 0 \\ \sin\psi_{g_i} & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{U} \quad (4)$$

where $\boldsymbol{\xi}_{g_i} = [x_{g_i}, y_{g_i}]^T$ denotes the position of the i^{th} ground robot in $\mathcal{I}$ and ψ_{g_i} its orientation. The linear and angular velocities are considered as control inputs of the vehicle, i.e., $\boldsymbol{U} = [v_{g_i}, \omega_{g_i}]^T$.

III. Cooperative Navigation Architecture

Our cooperative navigation architecture is composed of two main parts, one dedicated to aerial monitoring for targets' search and an other one for targets' reaching. Each part is decomposed in some subroutines to achieve the final goal of each main part. Each of them is described in the following.

A. Aerial monitoring

Three main tasks compose this part and they are performed only by the aerial drone; trajectory generation for scanning all the surface, the trajectory tracking and the target detection. The two last one are done at the same time. In this work, targets are represented by specific markers scattered in the environment.

1) Trajectory generation: To scan the area, a trajectory must first be designed using the shape and measurements of the surface, in order to cover the entire search area. To simplify further computations and streamline, we have chosen a rectangular area of size $L_x \times L_y$ centered in $\boldsymbol{p}_c = \begin{bmatrix} c_x \\ c_y \end{bmatrix}$, see Figure 3. Nevertheless, other types of surfaces can also be used for the trajectory design.

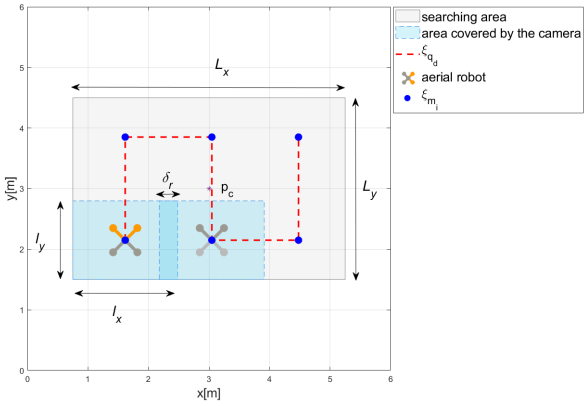

Fig. 3: Path-planning proposed to monitor the entire environment in a constant altitude.

Without loss of generality, the area covered by the camera can be modeled as a rectangle of size $l_x \times l_y$ where

$$l_x = \frac{z_q . r_x}{f . D_x} \quad l_y = \frac{z_q . r_y}{f . D_y}$$

with f represents the focal length, (r_x, r_y) the camera's resolution and D_x and D_y are the number of pixels per unit length along respectively x and y axis, in the camera frame. We consider that the drone flies at a desired altitude z_{q_d}, i.e. $z_q \to z_{q_d}$

Then, the waypoints (meeting points ξ_{m_i}) for the drone can be computed as

$$\boldsymbol{\xi}_{m_i} = \begin{cases} \boldsymbol{p}_c - \frac{1}{2} . \begin{pmatrix} L_x \\ L_y \end{pmatrix} + \frac{1}{2} . \begin{pmatrix} l_x(1+i) \\ l_y \end{pmatrix} - \frac{1}{2} . \begin{pmatrix} \delta_r . i \\ 0 \end{pmatrix} \\ \text{if i is even} \\ \boldsymbol{p}_c - \frac{1}{2} . \begin{pmatrix} L_x \\ -L_y \end{pmatrix} + \frac{1}{2} . \begin{pmatrix} l_x . i \\ -l_y \end{pmatrix} - \frac{1}{2} . \begin{pmatrix} \delta_r . (i-1) \\ 0 \end{pmatrix} \\ \text{else} \end{cases}$$

for $i : 0 : n$ with n is the number of waypoints required to scan the entire zone and $\delta_r > 0$ a small overlapping constant to make possible targets detection close to the border of the vision area. These points compose the desired trajectory to be tracked by the aerial drone.

2) Trajectory tracking: Once the meeting points ξ_{m_i} are computed, trapezoidal speed profile trajectories are generated. These trajectories are defined by desired points $\boldsymbol{\xi}_{q_d}(\xi_m, t) = [x_d(t), y_d(t), z_d]$ and velocities $\dot{\boldsymbol{\xi}}_{q_d}$ that will be used for the aerial vehicle controller.

The aerial drone's controller is given by

$$\begin{aligned} \boldsymbol{F}_u &= m_q \boldsymbol{g} - K_{d_t} \varsigma_{\beta_d} \left(\dot{\boldsymbol{\xi}}_e \right) - K_{p_t} \varsigma_{\beta_p} \left(K_{d_t} \boldsymbol{\xi}_e + \dot{\boldsymbol{\xi}}_e \right) \\ \boldsymbol{\tau}_q &= \boldsymbol{\Omega} \times J \boldsymbol{\Omega} - K_{d_\vartheta} \boldsymbol{\Omega}_e - K_{p_\vartheta} (2 \ln \boldsymbol{q}_e) \end{aligned} \tag{5}$$

where $K_{p_t}, K_{d_t}, K_{p_\vartheta}, K_{d_\vartheta} \in \mathbb{R}^{3\times 3}_+$ are constant gain matrices, $\boldsymbol{g} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$. In addition, $\boldsymbol{F}_u \to \boldsymbol{F}_{th}$ when $\boldsymbol{q} \to \boldsymbol{q}_d$. Moreover $\varsigma_\beta(\cdot)$ defines a saturation function defined as $\varsigma_\beta(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ for a vector $\vec{\lambda} \in \mathbb{R}^n$ and a positive constant scalar $\beta \in \mathbb{R}_+$ as

$$\varsigma_\beta \left(\vec{\lambda} \right) := \begin{cases} \vec{\lambda} & , \left\| \vec{\lambda} \right\| < \beta \\ \beta \, sgn \left(\vec{\lambda} \right) & , \left\| \vec{\lambda} \right\| \geq \beta \end{cases} \tag{6}$$

where the **sign** function $sgn(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ is defined element-wise. The stability analysis of this controller was demonstrated in [13]. This control algorithm will assure the position errors converge to zero, i.e. $\boldsymbol{\xi}_e \to 0$ and which implies that $\boldsymbol{\xi}_q \to \boldsymbol{\xi}_{q_d}$. Moreover, we will bound the desired velocity, $|\dot{\boldsymbol{\xi}}_{q_d}| \leq \epsilon$ with $\epsilon > 0$ small assuring that $z_q = z_{q_d}$ and $(x_q, y_q) \| (x, y)$.

During the trajectory tracking the target detection algorithm is being executed online and the position of the found targets is estimated $\hat{\boldsymbol{\xi}}_{T_j}$.

3) Targets' detection: ArUco markers are square-based fiducial markers with binary codes. We chose these markers as targets for easy practical implementation and because this method is robust against false detections.

We have improved work in [9] by proposing a robust candidate detection method, which takes into account the specific features of aerial images. In the cited method, a good separation of the marker from its environment is required and a dark object, even of small size, present on the margin of the marker prevents the detection. In our method, once the length of the marker edges, d, is known and with the fact that the controller ensures that the aerial drone is parallel to the ground, i.e., $(x_q, y_q) \| (x, y)$, then the marker to be detected in the image frame will be a square of size $a = f . \frac{d}{z_q}$, in pixels.

Our algorithm, for markers contour extraction and filtering can be summarized as follows:

1) *Binarization* of the image using an adaptive threshold.
2) Image *segmentation* by connected component *labeling*. This step is necessary to avoid the fusion of components during the morphological operations.
3) *Filtering* to remove too small connected components (area $<< a^2$).
4) *Morphological operations* taking into account the marker's size : these operations are a morphological closing operation with a disk shape structuring element of diameter $\frac{2.a}{b}$, followed by an opening operation with a disk shape structuring element of diameter $\frac{a}{b}$ being b the size of the marker in bits. Morphological operations are processed on each segmented component taken separately to prevent their fusion.
5) *Contour extraction* and *polynomial approximation*.
6) *Filtering* to get only square marker candidates of size a.

The advantage to use this algorithm is that it is possible to detect markers even in presence of an object of size

smaller than $\frac{a}{b}$ on the white margin of the marker.

The four corners' positions of the detected marker in the image frame $[u_i, v_i]^T$, $i : 1 : 4$ are used as inputs for the pose estimation algorithm. The relation between a 3D corner pose in the camera frame ${}^C\boldsymbol{\xi}_{Tc_j}$ and its 2D projection in the image frame is given by :

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \boldsymbol{A}.\boldsymbol{\Pi} \begin{bmatrix} {}^C\boldsymbol{R}_m & {}^C\hat{\boldsymbol{\xi}}_{T_j} \\ [0]_{3\times 1} & 1 \end{bmatrix} \begin{bmatrix} {}^m\boldsymbol{\xi}_{Tc_i} \\ 1 \end{bmatrix} \tag{7}$$

where ${}^C\boldsymbol{R}_m$ and ${}^C\hat{\boldsymbol{\xi}}_{T_j}$ represent the unknown rotation and translation vectors from the marker to the camera frame. $\boldsymbol{A}$ is the camera intrinsic parameters matrix and $\boldsymbol{\Pi}$ the perspective projection model. From those four points and using equation (7), the optimization function solvePnP [14] is able to estimate ${}^C\boldsymbol{R}_m$ and ${}^C\hat{\boldsymbol{\xi}}_{T_j}$ that minimize the retro-projection error.

The computed position is then projected in the inertial frame, $\mathcal{I}$, using the following relation

$$\begin{bmatrix} {}^{\mathcal{I}}\hat{\boldsymbol{\xi}}_{T_j} \\ 1 \end{bmatrix} = {}^{\mathcal{I}}\boldsymbol{T}_{\mathcal{B}}.{}^{\mathcal{B}}\boldsymbol{T}_C \begin{bmatrix} {}^C\hat{\boldsymbol{\xi}}_{T_j} \\ 1 \end{bmatrix} \tag{8}$$

Observe that the precision of $\hat{\boldsymbol{\xi}}_{T_j}$ in $\mathcal{I}$ is related with the camera parameters (intrinsic and extrinsic) and the detection quality (related with the camera's resolution, the marker size and the sharpness of the image).

The position of $\hat{\boldsymbol{\xi}}_{T_j}$ is then sent from the aerial drone to the ground station via a wifi protocol. This information is stored for beginning the allocation task.

B. Targets' reaching

Once the aerial vehicle has covered all the surface, it goes to the center of mass of the ground vehicles, for activating the targets' reaching task.

1) Allocation task: We have inspired our task allocation algorithm from the Auction Based Algorithm [10]. The advantage of our algorithm is that it is computed on-line and on-board of the aerial vehicle. It computes the task for each ground robot considering distance between each one of them to the nearest target. The allocation method is summarized in Algorithm 1.

Algorithm 1 Continuous task allocation algorithm

Require: List of robots and list of tasks
  Compute $\boldsymbol{D}$
  while D is not empty **do**
    Search minimal distance in $\boldsymbol{D}$
    $i \leftarrow$ index row of min($\boldsymbol{D}$)
    $j \leftarrow$ index column of min($\boldsymbol{D}$)
    Send the j^{th} task to the robot i^{th}
    Remove the j^{th} column and i^{th} row of $\boldsymbol{D}$
  end while

From Algorithm 1, $\boldsymbol{D}$ is the distance matrix of size $n \times m$ and is defined as

$$\boldsymbol{D} = (d_{i,j})_{1\leq i\leq n,\ 1\leq j\leq m}$$

with $d_{i,j} = \left\| \boldsymbol{\xi}_{g_i} - \hat{\boldsymbol{\xi}}_{T_j} \right\|$ representing the Euclidean distance between the robot i and the target j, n the number of robots and m the number of tasks.

Notice that this allocation algorithm is iterative and linked to the robots' position and reached tasks (targets). Therefore, it is necessary to update the whole allocation each time a robot reaches an assigned target.

2) Ground and aerial vehicles navigation: We propose a reactive local planner with an online collision inspired by the Dynamic Window Approach (DWA) strategy [11] to reach the target points. The objective function $G(v, \omega)$ has been modified to reduce the complexity of the original method. In the DWA, the goal is to choose at each time step a pair of angular and linear velocity among the admissible velocities that maximizes $G(v, \omega)$ defined in the following equation

$$\begin{aligned} G(v,\omega) &= \alpha.heading(v,\omega) + \delta.dist(v,\omega) \\ &\quad +\gamma.velocity(v,\omega) \end{aligned} \tag{9}$$

The function $heading$ quantifies how close is the robot to the goal, the function $dist$ quantifies how far it is from obstacles and the function $velocity$ quantifies whether the robot is close to the desired speed. The algorithm's complexity is $O(n \times m)$ with n and m the size of the search space of admissible velocities. This leads to find a compromise between the discretization step of the velocities and the execution time.

We have modified equation (9) by changing $heading(v, \omega)$, $dist(v, \omega)$ and $velocity(v, \omega)$ to convex loss function [15]. The new objective function $\mathcal{L}(v, \omega)$ defined in equation (10) is as well a convex function that must be minimize. The couple of linear and angular velocity that minimize $\mathcal{L}(v, \omega)$ can be obtained by applying a gradient descent on $\mathcal{L}(v, \omega)$.

$$\begin{aligned} \mathcal{L}(v,\omega) &= \alpha.heading_{loss}(v,\omega) + \delta.dist_{loss}(v,\omega) \\ &\quad +\gamma.velocity_{loss}(v,\omega) \end{aligned} \tag{10}$$

The original concept of search space is not modified. The gradient descent is limited in the search space. This modification enable us to reduce by four approximately the number of calculation.

IV. Experimental results

A. Experimental platform

The experimental setup consists of two ground robots Turtlebot3 equipped with a LiDAR to detect obstacles. They are programmed using ROS. The aerial drone is a quadcopter Parrot AR.Drone 2.0 equipped with a downward-facing camera running at 15 fps with a 320x240 pixels resolution. Its firmware has been modified to work with the open source software FL-AIR (Framework Libre Air) [12], used for its programming. The UAV communicates with

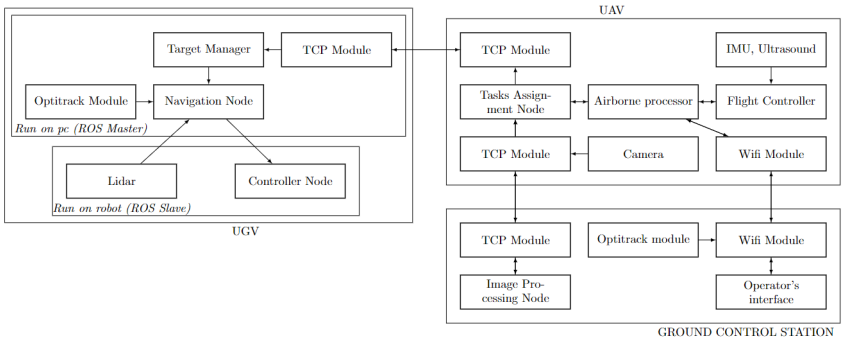

Fig. 4: Framework of the cooperative system. A module UGV is on-boarded on each ground robot.

the ground robots via a TCP socket. An OptiTrack motion capture system is used to estimate the robots' position at 100Hz with a precision of 1 mm.

The platform includes also a remote computer called Ground Control Station (GCS) used as human-uav interface. The GCS is connected to the Optitrack software and communicates robots' position and flight parameters to the UAV. It is also used to graph in real time the states of the UAV for analysis purposes. This remote computer is also responsible for processing the drone images that are sent via TCP sockets. Figure 4 summarizes the proposed framework.

Figure 5 illustrates the experimental environment. Five ArUco markers of size 10 cm $\times$ 10 cm have been placed randomly in the search area. Each marker encodes a unique identifier facilitating the position estimation merging. The final estimation of the j^{th} marker's position $\overline{\hat{\boldsymbol{\xi}}_{T_j}}$ is the mean of the position estimations $\hat{\boldsymbol{\xi}}_{T_j}$. Unmapped obstacles have been placed across the way of the UGVs.

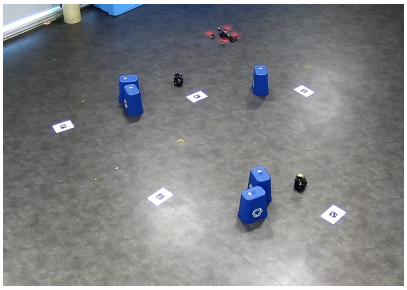

Fig. 5: Experimental environment.

B. Results

Experimental tests in real time were conducted to validate the proposed architecture.

1) Aerial monitoring and targets' searching: The goal is to verify the well performance of the control and vision algorithm. For the vision algorithm, we use markers with a blank margin (Figure 6) and we add black splines in the markers's margin for perturbing the algorithm.

In Table I, the accuracy of the marker position estimation algorithm is presented. We note that all the 5 markers were detected by using our method. When using the original detection algorithm, in [9], we observed that the markers detection was impossible. During the aerial monitoring, the drone

Fig. 6: Example of markers with black splines in their margin used for the tests.

follows, with pretty good precision, the desired trajectory scanning the entire area, see Figure 7 and 9. At the end of the aerial monitoring task, all markers are detected with an estimation error smaller than 10 cm. This error is mainly due to the estimation of the camera's intrinsic parameters. In fact, we observed that the most accurate markers' placement estimations are those positioned close to the scan path. We have also observed that the height of the drone and the camera resolution also impacts the results: for $z > 2$m, the marker is no longer readable.

Presence of an object in the markers' margin	Markers detected	Mean pose estimation error (mm)
No	5/5	85
Yes	5/5	78

TABLE I: Markers' detection and pose estimation results during one scan of the search area.

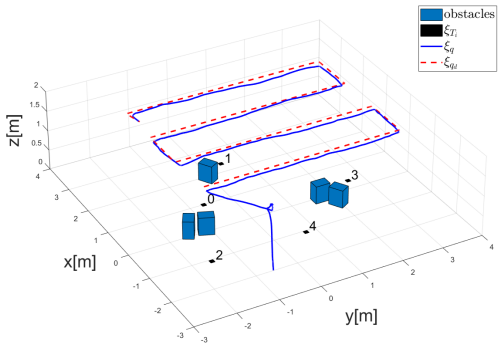

Fig. 7: Real-time results when the UAV follows a desired trajectory during searching mission for finding targets.

2) Targets' reaching: To verify the target locations, two UGV must navigate to their nearest target avoiding obstacles, while the UAV monitors the ground mission performance by flying in the center of mass of both vehicles.

In Figure 8 robots' trajectories during the targets-reaching step are shown. As expected, ground robots reach targets in a straight line. However, their trajectories become curved to avoid obstacles. We can also observe that the drone tracks the motion of the robots well. When the ground vehicle is into a neighborhood of the target (previously defined), the drone considers that the target is reached and the tasks are then immediately updated.

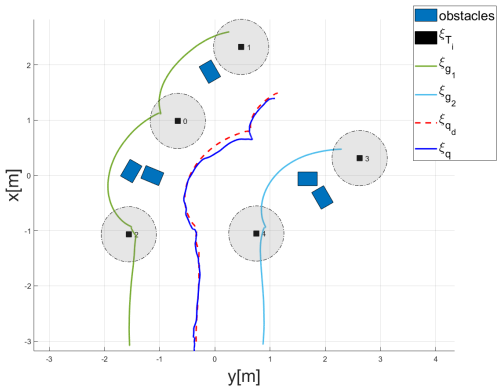

Fig. 8: Robots trajectories during the targets reaching task.

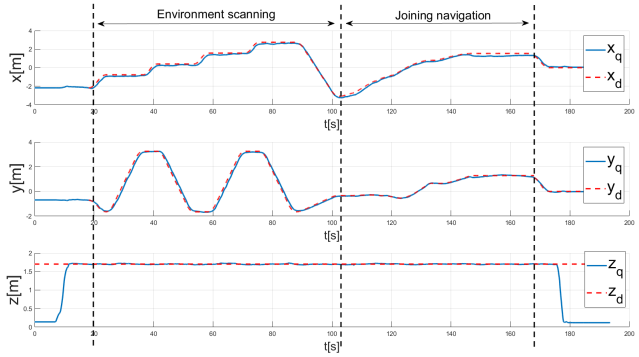

Fig. 9: UAV trajectory performance along x, y and z axis

Notice from Figures 5 - 9 the good performance of the proposed architecture for searching and reaching unknown targets. A video of this experiment can be see at : https://youtu.be/Mu-w_kA5dNw.

V. Conclusion and Future Works

In this paper, a centralized cooperative architecture for searching and reaching unknown targets was proposed. The whole robotic system is composed of one aerial vehicle and two ground vehicles. An aerial target detection algorithm was first developed to estimate the relative position of the unknown targets. This information was then used to the cooperative architecture to assign tasks to the ground vehicles in order to reach their closest target. The robots navigation was done in autonomous mode and the ground vehicle control was capable to avoid obstacles. Real-time experiments corroborated the well performance of the system.

This solution can potentially be used as a basis for target recognition applications. Further works will be take into account communication and localization hazards to increase robustness. Possible extensions in dynamic environments will be studied and will require to improve the cooperation between robots to make the system more flexible and adaptable.

References

[1] C. Chen, Y. Wan, B. Li, C. Wang, G. Xie and H. Jiang. (2020). "Motion Planning for Heterogeneous Unmanned Systems under Partial Observation from UAV". In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA, pp. 1474-1479, Oct 2020.

[2] Z. Su, C. Wang, X. Wu, Y. Dong, J. Ni and B. He. (2021). "A Framework of Cooperative UAV-UGV System for Target Tracking". In 2021 IEEE International Conference on Real-time Computing and Robotics (RCAR), Xining, China, pp. 1260-1265, Jul 2021.

[3] H. Qin, Z. Meng, W. Meng, X. Chen, H. Sun, F. Lin and M. H. Ang Fr. (2019). "Autonomous Exploration and Mapping System Using Heterogeneous UAVs and UGVs in GPS-denied Environments". In IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1339-1350.

[4] A. Lakas, B. Belkhouche, O. Benkraouda, A. Shuaib and H. Alasmawi. (2018). "A Framework for a Cooperative UAV-UGV System for Path Discovery and Planning". In 2018 International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, pp. 42-46, Nov 2018.

[5] Wang, L. et al. (2020). "A Collaborative Aerial-Ground Robotic System for Fast Exploration". In: Xiao, J., Kröger, T., Khatib, O. (eds) Proceedings of the 2018 International Symposium on Experimental Robotics. ISER 2018. Springer Proceedings in Advanced Robotics, vol 11. Springer, Cham.

[6] I. D. Miller, F. Cladera, T. Smith, C. J. Taylor and V. Kumar. (2022). "Stronger Together: Air-Ground Robotic Collaboration Using Semantics". In IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 9643-9650.

[7] C. Shen, Y. Zhang, Z. Li, F. Gao and S. Shen. (2017). "Collaborative Air-Ground Target Searching in Complex Environments". In 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, pp. 230-237, 11-13 October 2017.https://fr.overleaf.com/project/633db0c375b9ae0730e4ae70

[8] P. Castillo, R. Lozano and A. Dzul. (2004). "Stabilization of a mini-rotorcraft having four rotors". In 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, vol.3, pp. 2693-2698, Sept 2004.

[9] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, M.J. Marín-Jiménez. (2014). "Automatic generation and detection of highly reliable fiducial markers under occlusion". In Pattern Recognition, vol. 47, Issue 6, pp. 2280-2292.

[10] G. Lozenguez, L. Adouane, A. Beynier, A. Mouaddib, P. Martinet. (2016). "Punctual versus continuous auction coordination for multi-robot and multi-task topological navigation". In Autonomous Robots, Springer Verlag, vol. 40, no. 4, pp.599-613.

[11] D. Fox, W. Burgard, and S. Thrun. (1997). "The dynamic window approach to collision avoidance". In IEEE Robotics and Automation Magazine, vol. 4, no. 1, pp. 23–33.

[12] G. Sanahuja et al, «Fl-AIR - Framework libre AIR,» Heudiasyc, 2012. [Online]. Available: https://devel.hds.utc.fr/software/flair. [accessed 21-07-2022].

[13] J. Carino, H. Abaunza, P. Castillo, (2022), "A Fully-Actuated Quadcopter Representation using Quaternions", TCON, International Journal of Control, 2022. https://doi.org/10.1080/00207179.2022.2129789

[14] E. Marchand, H. Uchiyama and F. Spindler. (2016). "Pose Estimation for Augmented Reality: A Hands-On Survey". In IEEE Transactions on Visualization and Computer Graphics, vol. 22, no. 12, pp. 2633-2651.

[15] H. Pousseur, A. Victorino; "Gradient Descent Dynamic Window Approach to the Mobile Robot Autonomous Navigation". IEEJ International Workshop on Sensing, Actuation, Motion Control, and Optimization (SAMCON'2022), Saitama, Tokyo, March 2022.