A Decentralized Interactive Architecture for Aerial and Ground Mobile Robots Cooperation

El Houssein Chouaib Harik, François Guérin, Frédéric Guinand, Jean-François Brethé, Hervé Pelvillain

To cite this version:


HAL Id: hal-01145928
https://hal.archives-ouvertes.fr/hal-01145928
Submitted on 27 Apr 2015

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.

Copyright
A Decentralized Interactive Architecture for Aerial and Ground Mobile Robots

El Houssein Chouaib Harik
LITIS
University of Le Havre
Le Havre, France
el-houssein-chouaib.harik@univ-lehavre.fr

François Guérin
GREAH
University of Le Havre
Le Havre, France
francois.guerin@univ-lehavre.fr

Frédéric Guinand
LITIS
University of Le Havre
Le Havre, France
frederic.guinand@univ-lehavre.fr

Jean-François Brethé
GREAH
University of Le Havre
Le Havre, France
jean-francois.bretche@univ-lehavre.fr

Hervé Pelvillain
IUT
University of Le Havre
Le Havre, France
herve.pelvillain@univ-lehavre.fr

Abstract—This paper presents a novel decentralized interactive architecture for aerial and ground mobile robots cooperation. The aerial mobile robot is used to provide a global coverage during an area inspection, while the ground mobile robot is used to provide a local coverage of ground features. We include a human-in-the-loop to provide waypoints for the ground mobile robot to progress safely in the inspected area. The aerial mobile robot follows continuously the ground mobile robot in order to always keep it in its coverage view.

Keywords—Unmanned Aerial Vehicles, Unmanned Ground Vehicles, Air-Ground-Cooperation, non-linear kinematic controller, DES algorithm.

I. INTRODUCTION

Unmanned Aerial and Ground Vehicles (UAVs, UGVs) cooperation attracts increasingly the attention of researchers, essentially for the complementary skills provided by each type to overcome the specific limitations of the others. UAVs provide a global coverage and faster velocities, and UGVs provide a higher payload and stronger calculation capabilities, adding a local coverage for the unseen areas from an aerial perspective. In addition, deploying both types together ensure faster and more reliable results within a shorter frame of time compared to the deployment of a single type of mobile robots.

Air-Ground-Cooperation (AGC) in mobile robots systems can envisage a large panel of applications. Our focus in this paper is on Surveillance or Inspection (SI) missions. The authors in [1], [2],[3] use a Decentralized Data Fusion (DDF) technique [4] for exploring a specific area. The task of both air and ground based nodes is to make observations of terrain features and identify moving or stationary targets.

The authors in [5] use AGC for target searching missions. Their testbed is composed of three layers: a high altitude UAV (a planner) to determine the motion of the mobile robots based on the desired goals, medium altitude UAVs (Blimps) to track the group of Unmanned Ground Vehicles (UGVs) and try to maintain them in their field of view, and the UGVs that navigates for exploration or other tasks. Another interesting work can be found in [6], where the authors present a system that allow to deploy a significant number of ground mobile robots monitored by aerial mobile robots (aerial shepherds) for target searching. An urban environment surveillance application can be found in [7]. The authors of this work cover a large set of technological topics from mapping to communication and collaborative navigation using multiple aerial and ground mobile robots. The authors in [8] also use a group of UAVs to provide an aerial coverage to a group of UGVs used to clear a path through that area. We can find in [9] a surveillance scheme including a group of six UGVs and four UAVs. An interagent cohesion and separation and a velocity synchronization scheme were combined into a control and communication strategy for mobile robots navigation. Another work of the same authors [10] present an AGC scheme for target detection mission. The size of the navigation area is related to the number of the deployed mobile robots. A set of grid points is defined for UGVs. An interagent potential force based approach is used for the UGVs navigation towards the grid points. Once the UGVs reached their grid points the aerial mobile robots start scanning the field. The authors in [11], [12] use a UAV as a remote sensor that flies ahead the UGV to provide geo-referenced 3D geometry, in order to help the UGV for navigating in the area avoiding obstacles.

In the previous related works, a first flight of the drone over the inspected area is always performed before the beginning of the mission. A traversability map is then processed to provide a trajectory to the UGV. The main contribution of our work consists in providing a real-time navigation scheme: the UAV flies over the area to provide a global coverage, and to assist a UGV in real-time to navigate safely avoiding obstacles. To provide the real-time navigation scheme without the need to have a per-processed map, a human operator is used to provide waypoints by monitoring the global and local coverages, making the architecture interactive, and applicable to real scenarios. Indeed, for numerous reasons, monitoring, surveillance and inspection of industrial sites belonging to Seveso category (highly critical sites) cannot be performed without a constant and careful attention of human experts.
The paper is organized as follows: In section II we introduce the problem statement and define the objectives of our work. Section III presents the UGV and UAV controllers design. We discuss in section IV and V the simulation and experimental results, and we conclude in section VI the present work and discuss future perspectives.

II. PROBLEM STATEMENT

A UAV is used to monitor a given area where a UGV is present to perform the inspection task. The images taken from the camera mounted on the UAV are sent continuously to the ground station. The images are processed in real time to provide localization information to the UGV to navigate in the area through the on-the-go waypoints selected by a human operator (Figures 1 and 2).

To successfully inspect a given area within a reasonable frame of time, global and local coverages must be acquired simultaneously, which can be possible if we deploy both types of mobile robots (aerial and ground) simultaneously. In this direction, our architecture is composed of a UAV equipped with a down-facing camera, and a UGV equipped with a horizontal-facing camera. Both video flows are sent continuously to a ground station running our developed Human-Machine-Interface (HMI). A human operator supervises the video flows, and selects progressively navigation waypoints for the UGV. As the UGV navigates through the given waypoints, the UAV follows continuously the UGV in order to keep it in its coverage view (the center of the image plan) using visual servoing.

III. UGV AND UAV CONTROLLERS DESIGN

A. Hardware configuration

We consider in this work a non-holonomic, unicycle-like, Wheeled Mobile Robot (WMR) that has been designed in the GREAH laboratory of the University of Le Havre (France) [13]. The UAV is a quadcopter (Phantom 2 Vision) developed by DJI [14] with a sufficient payload to carry an open platform device (android phone) that is used to take the video flow and send it to the ground station continuously via Wifi (IEEE 802.11). The received flow is processed to extract the position and orientation (pose estimation) of the UGV relative to the selected waypoint.

B. Pose estimation of the UGV

To successfully navigate in a given area, the relative pose of the UGV should be known (orientation and location). The authors of AGC applications ([15], [16], [17], [18]) used a single colored marker to define the relative location of the UGV to the UAV, and a digital compass to communicate the orientation which generally gives inaccurate measurements (digital compass is sensible to electromagnetic field variations). To overcome this issue, we added a second marker to also estimate its orientation.

We consider $X_c$ and $Y_c$ the axis of the image (camera) taken from the UAV (Figure 3). In order to locate the UGV, we used a color tracking algorithm to extract the position of colored markers: $R_c$ (red marker) fixed on the center of the driving wheels, and $R_h$ (blue marker) fixed on the head of the UGV. The position (in pixels) of $R_c$ and $R_h$ along $X_c$ and $Y_c$ axis are respectively: $Y_{R_c}$, $X_{R_c}$ and $Y_{R_h}$, $X_{R_h}$. We assume that the center of the camera (center of the image) is the center of the UAV, thus we assume that the camera frame ($X_c$, $Y_c$) and the UAV inertial frame ($X_D$, $Y_D$) are superimposed.

The orientation $\alpha$ of the UGV to a given waypoint $w$, can be computed as $\alpha = \theta - \beta$ with:

$$\theta = \arctan\left(\frac{Y_w - Y_{R_c}}{X_w - X_{R_c}}\right) \quad \beta = \arctan\left(\frac{Y_{R_h} - Y_{R_c}}{X_{R_h} - X_{R_c}}\right)$$

(1)

Note that by using the colored markers on the UGV, we can extract its orientation without knowing the orientations of the UAV and the UGV in the world frame. The distance $d$ that...
separates the UGV and the waypoint \( w \) has to be expressed in the world frame. It can be written as follows:

\[
d = \sqrt{(x_w - x_Re)^2 + (y_w - y_Re)^2}
\]  

(2)

Where \( x_w, y_w, x_Re, y_Re \) represent (in meter) the coordinates of the \( R_e \) marker and the waypoint in the UAV or camera frame previously supposed to be the same. They can be computed using the classical Pinhole model of the camera [19]. The example is given for the waypoint:

\[
\begin{bmatrix}
X_w \ z \\
Y_w \ z \\
\end{bmatrix} = 
\begin{bmatrix}
G_x & 0 & 0 & z.X_0 \\
0 & G_y & 0 & z.Y_0 \\
\end{bmatrix} \begin{bmatrix}
x_w \\
y_w \\
z \\
\end{bmatrix}
\]

(3)

Where:

- \( f \): Focal distance - \( \Delta_x, \Delta_y \): Dimensions of a pixel.
- \( X_0, Y_0 \): Projection (in pixels) of the camera optical center in the image plan. They are assumed to be null.
- \( X_w, Y_w \): Projection (in pixels) of the waypoint in the image plan.

We note as well that using the pinhole model we can estimate the altitude of the UAV using the distance between the two colored markers (red and blue).

C. Design of the UGVs controller

The UGV receives \( d \) and \( \alpha \) via a radio module from the ground station. The goal is to use these information to generate \( u \) and \( r \) that respectively represent the longitudinal and angular velocities of the UGV (Figure 4). The control objective is to regulate the distance \( d \) and the steering angle \( \alpha \) as follows:

\[
\lim_{t \to \infty} d(t) = L \quad \lim_{t \to \infty} \alpha(t) = 0
\]  

(4)

The time derivative of the distance errors (5), which depends on both the UGV’s velocities \( (u, r) \) and the target’s velocities along the \( X_R \) and \( Y_R \) axes, denoted as \( TV_{X_R} \) and \( TV_{Y_R} \), are given by:

\[
\dot{e}_{X_R} = d.\cos(\alpha) - \dot{d}.\alpha.\sin(\alpha) = TV_{X_R} - u \\
\dot{e}_{Y_R} = d.\sin(\alpha) + \dot{d}.\alpha.\cos(\alpha) = TV_{Y_R} - r L
\]  

(6)

Where \( rL \) corresponds to the linear velocity of \( R_h \) (blue marker).

\( TV_{X_R} \) and \( TV_{Y_R} \) are null since the target of the UGV is a waypoint. We propose then the following error dynamic equations:

\[
\dot{e}_{X_R} = -K.e_{X_R} = -u \\
\dot{e}_{Y_R} = -K.e_{Y_R} = -r.L
\]  

(7)

Where \( K > 0 \) is the proportional gain of the UGV’s controller. It is chosen to get a first order dynamic behavior for the control objective:

\[
\dot{e}_{X_R} + K.e_{X_R} = 0 \\
\dot{e}_{Y_R} + K.e_{Y_R} = 0
\]  

(8)

The numerical value given to \( K \) allows to define the decreasing speed of the distance errors \( e_{X_R} \) and \( e_{Y_R} \). Note that for large values of the distance \( d \), a saturation is implemented to consider the limitations of the electrical drives that control the UGV’s left and right wheels.

Starting from (7) we propose the following non-linear kinematic controller:

\[
u = K.(d.\cos(\alpha) - L) \\
r = K.(d.\sin(\alpha))/L
\]  

(9)

D. Design of the UAV controller

To hover the UAV over the UGV during its waypoints navigation, we need to control the UAV movements along \( X_D \) and \( Y_D \) axis (Figure 3) since the UGV moves on a 2D plan (assumed to be flat). We need to develop a navigation controller that takes as input the location of the UGV \( (R_e) \), and generates the necessary pitch and roll angles to keep the UGV at the center of the image plan (visual servoing).

We consider four inputs \( \phi_d, \theta_d, \psi_d \) and \( z_d \). They represent respectively the desired roll, pitch, yaw angles, and the desired altitude. Note that the UAV attitude (roll, pitch, yaw, altitude) is controlled by the internal autopilot. We suppose that the UAV flies at a fixed altitude, and a fixed yaw: \( z_d = z \) and \( \psi_d = 0 \). \( z \) is chosen to have a sufficient coverage (3) of the inspected area \( (z_{min} < z) \), and to keep the track of the UGV \( (z < z_{max}) \). \( z_{min} \) and \( z_{max} \) can be defined experimentally.

To design the UAV controller, we used the dynamic model of a quadcopter [20]. Starting from this dynamic model, the movements along \( X_D \) and \( Y_D \) axis can be described as follows:

...
\[\dot{x}_D = (\cos \phi_d \sin \theta_d \cos \psi_d + \sin \phi_d \sin \psi_d) \frac{1}{m} U_1 \]
\[\dot{y}_D = (\cos \phi_d \sin \theta_d \sin \psi_d - \sin \phi_d \cos \psi_d) \frac{1}{m} U_1 \]

Equations (10) can be as:
\[\ddot{x} = u_{X_D} \frac{1}{m} U_1 \]
\[\ddot{y} = u_{Y_D} \frac{1}{m} U_1 \]

With:
\[u_{X_D} = (\cos \phi_d \sin \theta_d \cos \psi_d + \sin \phi_d \sin \psi_d) \]
\[u_{Y_D} = (\cos \phi_d \sin \theta_d \sin \psi_d - \sin \phi_d \cos \psi_d) \]

where \(u_{X_D}, u_{Y_D}\) represent the orientations of the total thrust \((U_1)\) responsible for linear motions of the quadcopter along \(X_D\) and \(Y_D\) axis. \(m\) is the UAVs weight.

To allow the UAV to keep the UGV in its coverage view (the center of the image plan), let us consider the following distance errors (Figure 3):
\[e_{X_D} = x_{Re} - X_0 \quad e_{Y_D} = Y_{Re} - Y_0 \]

Note that \(X_0, Y_0\) are assumed to be null. The control objective is to regulate the distance errors as follows:
\[\lim_{t \to \infty} e_{X_D}(t) = 0 \quad \lim_{t \to \infty} e_{Y_D}(t) = 0 \]

To fulfill this control objective, we propose the following dynamic equations:
\[\ddot{e}_{X_D} = -K_{1X_D} e_{X_D} - K_{2X_D} \dot{e}_{X_D} = T_{AXD} - \ddot{x}_D \]
\[\ddot{e}_{Y_D} = -K_{1Y_D} e_{Y_D} - K_{2Y_D} \dot{e}_{Y_D} = T_{AYD} - \ddot{y}_D \]

Where \(K_{1,2}(X_D,Y_D)\) are positive gains. They are chosen to get a second order dynamic behavior without damping and oscillations for the control objective:
\[\ddot{e}_{X_D} + K_{2X_D} \dot{e}_{X_D} + K_{1X_D} e_{X_D} = 0 \]
\[\ddot{e}_{Y_D} + K_{2Y_D} \dot{e}_{Y_D} + K_{1Y_D} e_{Y_D} = 0 \]

\(T_{AXD}\) and \(T_{AYD}\) are the accelerations of the UGV along \(X_D, Y_D\) axis. They are neglected compared to \(\ddot{x}_D\) and \(\ddot{y}_D\) since the UAV moves much faster than the UGV. Starting from (15), we propose the following controller:
\[u_{X_D} = \frac{m}{U_1} (K_1 e_{X_D} + K_2 \dot{e}_{X_D}) \]
\[u_{Y_D} = \frac{m}{U_1} (K_1 e_{Y_D} + K_2 \dot{e}_{Y_D}) \]

To find the desired roll and pitch angles needed to track the UGV, we combine (17) in (12) taking a fixed yaw angle \(\psi = 0\), the desired angles are then:
\[\phi_d = \sin(-w_{Y_D}) \]
\[\theta_d = \sin\left(\frac{u_{X_D}}{\cos \phi_d} \right) \]

We note that the steady state error will be null (when the UGV stops) since it is a position control.

E. DES algorithm

Furthermore, the Double Exponential Smoothing (DES) algorithm [21] has been implemented to smoothen the uncertain measurements given by the vision system. The DES algorithm runs approximately 135 times faster with equivalent prediction performances and simpler implementations compared to a Kalman filter. The DES algorithm at time instant \(n, T_e\), where \(T_e\) is the sampling period and \(n\) is the discrete-time index, implemented for the distance \(d\), the steering angle \(\alpha\), and the projection of the red marker \(X_{Re}, Y_{Re}\) is given as follows:
\[S_{m_n} = \gamma_n m_n + (1 - \gamma_n)(S_{m_{n-1}} + b_{m_{n-1}}) \]
\[b_{m_n} = \lambda_n (S_{m_n} - S_{m_{n-1}}) + (1 - \lambda_n)b_{m_{n-1}} \]

Where:
\(m_n\) is the value of \((d, \alpha, X_{Re}, Y_{Re})\) at \(n\)th sample instant.
\(S_{m_n}\) is the smoothed values of \((d, \alpha, X_{Re}, Y_{Re})\).
\(b_{m_n}\) is the trend value of \((d, \alpha, X_{Re}, Y_{Re})\).

Equation (19) smoothes the value of the sequence of measurements by taking into account the trend, whilst (20) smooths and updates the trend.

The initial values given to \(S_{m_n}, b_{m_n}\) are:
\[S_{m_1} = m_1 \quad b_{m_1} = m_2 - m_1 \]

Usually, \(\gamma\) \((0 \leq \gamma \leq 1)\) is called the data smoothing factor and \(\lambda\) \((0 \leq \lambda \leq 1)\) is called the trend smoothing factor. A compromise has to be found for the values of \(\gamma\) and \(\lambda\). High values make the DES algorithm follow the trend more accurately whilst small values make it generate smoother results. The smoothed values are used instead of the direct noisy measurements in the proposed controllers. DES algorithm has been used successfully in a previous work [13] for vision based target tracking.

IV. UGV Results

We will present in this section simulation and experimental results concerning only the UGV, the UAV is piloted by a professional.
A. Simulation results

We simulated a real-like scheme to navigate in a given area avoiding obstacles. For the UGV we have used the dynamic model fully described in [13] and [19]. Simulations have been carried out with Matlab-Simulink software (sampling period: 10 ms). For the UGV non-linear kinematic controller we used the following parameters: \( L = 15 \text{cm}, K = 0.1 \).

Figure 9 shows that the UGV is able to follow the given waypoints, which validates the proposed controller (9). The results concerning the control objective (4) have been checked experimentally.

B. Experimental results

We created a user interface running on the ground station using Processing [22]. The human operator can supervise in real time both video flows received from UAV and UGV (Figure 5).

The human operator can select waypoints in the image (global view side) by a simple click. As explained in section (III-B), the software extracts and sends the distance \( d \) and the steering angle \( \alpha \) to the UGV via ZigBee module (IEEE 802.15.4) at a frequency of 50Hz (real time interrupt).

The experiments were carried out with a smartphone mounted under the quadcopter. For safety reasons, the UAV was piloted by a professional, but we carried out as well other experiments without a pilot using an AR Drone 2.0. Videos can be found in [23] for the experiment with a pilot, and [24] for the newer version without a pilot.

![Figure 5. Aerial and ground perception on the HMI. Left: The global view from the UAV. Right: the local view from the UGV.](image)

Figure 6 illustrate the UGV experimentation: The pilot (1) manipulates the UAV (2) that takes and send to the ground station a video flow (3). The operator (4) selects new waypoints to guide the UGV (5).

Figure 6. Waypoint tracking experimentation.

![Figure 6. Waypoint tracking experimentation.](image)

V. UAV results

We present in this section only simulation results of the UAV following the UGV in its waypoints tracking because we do not yet have access to the UAV’s position measurements in the inertial frame (for example VICON MX system). That is why quantitative results are presented for the simulation trials only. Please refer to the links mentioned in section IV for experiments videos.

A. Simulation result

We used for simulation the parameters (table I) of an open-source quadcopter developed by [25]. The parameters have been obtained thanks to experimentations fully described in [26].

<table>
<thead>
<tr>
<th>parameter</th>
<th>significance</th>
<th>values</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>mass</td>
<td>1.4 Kg</td>
</tr>
<tr>
<td>( b )</td>
<td>thrust coefficient</td>
<td>1.3e5 Ns²</td>
</tr>
<tr>
<td>( d )</td>
<td>drag coefficient</td>
<td>1e9 Nms²</td>
</tr>
<tr>
<td>( J_r )</td>
<td>rotor inertia</td>
<td>6e-7 kg.m²</td>
</tr>
<tr>
<td>( L )</td>
<td>arm length</td>
<td>1 m</td>
</tr>
<tr>
<td>( I_x )</td>
<td>inertia on x axis</td>
<td>0.02582 kg.m²</td>
</tr>
<tr>
<td>( I_y )</td>
<td>inertia on y axis</td>
<td>0.02616 kg.m²</td>
</tr>
<tr>
<td>( I_z )</td>
<td>inertia on z axis</td>
<td>0.04543 kg.m²</td>
</tr>
<tr>
<td>( K_1 )</td>
<td>Positive gain 1</td>
<td>1.5</td>
</tr>
<tr>
<td>( K_2 )</td>
<td>Positive gain 2</td>
<td>3</td>
</tr>
</tbody>
</table>

![Figure 7. Distance variation](image)

Figure 7. Distance variation

![Figure 8. Steering angle variation](image)

Figure 8. Steering angle variation
work goes towards implementing an autonomous waypoints selection through free-space navigation extraction, which allow the testbed to be extended in more than one UGV, and let the human operator focus on other coordination tasks.

ACKNOWLEDGMENTS

The authors would like to thank Le Havre Town Council CODAH for their support under research grants, and "DroneXTR" for piloting the drone. They would like as well to thank Upper Normandie Region through their support to PCMAI project.

REFERENCES


[24] Experimentation 2: Autonomous flight, 2015. URL: https://www.youtube.com/watch?v=p2JAUmRrTtM.
