



HAL
open science

Fuzzy logic controller for predictive vision-based target tracking with an unmanned aerial vehicle

El Houssein Chouaib Harik, François Guerin, El Houssein, Chouaib Harik, François Guerin, Frédéric Guinand, Jean-François Brethé, Hervé Pelvillain, Jean-Yves Parédé

► To cite this version:

El Houssein Chouaib Harik, François Guerin, El Houssein, Chouaib Harik, François Guerin, et al.. Fuzzy logic controller for predictive vision-based target tracking with an unmanned aerial vehicle. *Advanced Robotics*, 2017, 31 (7), pp.368-381. 10.1080/01691864.2016.1271500 . hal-01958989

HAL Id: hal-01958989

<https://hal.science/hal-01958989>

Submitted on 11 Mar 2020

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.

Fuzzy Logic Controller For Predictive Vision Based Target Tracking With An Unmanned Aerial Vehicle

El Houssein Chouaib Harik^{a*}, François Guérin^b, Frédéric Guinand^c Jean-François Brethé^b, Hervé Pelvillain^d, and Jean-Yves Parédé^b

^a*LITIS, University of Le Havre, France.*; ^b*LITIS, University of Le Havre, France, and UKSW, University Cardinal Stefan Wyszyński in Warsaw, Poland.*; ^c*GREAH, University of Le Havre, France.*;

^d*IUT GEII, University of Le Havre, France.*

(v1.0 released April 2016)

We present in this paper a Fuzzy Logic Controller (FLC) combined with a predictive algorithm to track an Unmanned Ground Vehicle (UGV), using an Unmanned Aerial Vehicle (UAV). The UAV is equipped with a down facing camera. The video flow is sent continuously to a ground station to be processed in order to extract the location of the UGV and send the commands back to the UAV to follow autonomously the UGV. To emulate an experienced UAVs pilot, we propose a fuzzy-logic sets of rules. Double Exponential Smoothing Algorithm (DES) is used to filter the measurements and give the predictive value of the UGV pose. The FLC inputs are the filtered UGV position in the image plan and the derivative of its predicted value. The outputs are pitch and roll commands to be sent to the UAV. We show the efficiency of the proposed controller experimentally, and discuss the improvement of the tracking results compared to our previous work.

Keywords: Fuzzy logic sets, Unmanned Aerial Vehicle, Vision based target tracking, Augmented reality

1. Introduction

The usage of Unmanned Aerial Vehicles (UAVs) has known an exponential growth in the past years. This is mainly due to the democratization of their employment after being limited to military applications. Many research laboratories focus their activities on modeling, control, or final user applications. Compared to fixed wings, rotary wings UAVs cover the biggest part in research projects. The main advantage of rotary wings over fixed wings are the vertical takeoff and landing capabilities, and their small size which allow them to be deployed in indoor environment, but nevertheless, they suffer from limited range of applicability due to power consumption, and their limited payload.

When deployed, UAVs play generally the role of flying eyes [1]. For this purpose, they are equipped with perception sensors such as RGB or RGBD cameras, lidars, or other sensors. Their usage in this case is to provide global coverage of a given area. The collected data are processed to extract useful information of that area. These informations can be used to define waypoints for UGVs, to determine an optimal navigation path [2], or to extract of points of interests [3]. When equipped with depth cameras or lidars, the UAVs can be used to build a 3D maps of buildings [4], and a concrete example of this usage is illustrated in estimating the degree of damage inside a building after an earthquake [5].

*Corresponding author. Email: el-houssein-chouaib.harik@univ-lehavre.fr

The main lacuna in using UAVs as an aerial sensor is that ground features are omitted, which makes the gathered information incomplete. To acquire the unseen features, researchers deploy Unmanned Ground Vehicles (UGVs) alongside UAVs. The UGVs can be used as landing pads for the UAVs (rotatory wings) [6], and the same pad can serve for charging the batteries of the UAV [7]. Deploying both UAVs and UGVs in the same mission requires mutual localization, whether estimating the relative pose of the UGV to the UAV [8, 9], or the reverse [10], or mutual pose estimation when using GPS information.

In urban environments, GPS information is inaccurate, and sometimes inexistent. Thus researchers do not rely on this type of sensor for the mutual localization, and use instead vision based methods. They consist in recognizing visually special features on the UGV or the UAV. When estimating the position of the UGV using a camera mounted on the UAV, color detection is the first technique that has been used. It consists in placing a distinguished colored marker on the planar surface of the UGV [11–14], and to go through the acquired video from the UAV until a match is found with the predefined tracked color. Once the location of the tracked object is found, relative pose of the UGV is extracted. The advantages of the color tracking method is the easiness of implementation, the minimalistic processing time, and the good results in adequate lightning conditions. Nevertheless, there are some drawbacks. This method is sensitive to light changes. Going from a light spot to a dark one changes the tracked color intensity, resulting in the loss of the target. Another drawback is the difficulty to ensure that the surrounding environment does not have the same tracked color. This will result in a false detection, thus false estimation of the tracked targets pose.

To overcome the limitations of the color tracking method, some works got inspired by an Augmented Reality (AR) application [15], where the authors presented a Computer Supported Collaborative Work (CSCW). A conference system where remote collaborators are represented on virtual monitors which can be freely positioned in a user space [15]. The user is wearing a Head Mounted Display (HMD), and can see the collaborators faces overlaid on a specific marker. The proposed method is publicly available and known as the ARToolKit. The library has been used in [16], where the authors included it as a vision feedback for pose estimation in the control loop of several UGVs. We can find in [17] another interesting feature-based (black and white pattern) detection for multiple robots localization. The presented method allows the tracking of several features at a time with a good precision using low cost cameras, and the same kind of visual marker as in ARToolKit. The proposed method has been made publicly available as a library called ArUco [18]. Another AR marker tracking can be found in [19]. The authors used a down facing camera mounted on an UAV (quadrotor) in order to hover it on a ground feature (AR marker). They fused the information gathered from the vision combined with internal sensors (Inertial Measurement Unit (IMU)) in order to generate the necessary movements to stay on the top of the ground feature. In addition to track a moving target, the authors in [20] present a landing algorithm using an AR marker on the planar surface of the target. After patrolling the area, once the AR marker is found, the tracking process begins. The error between the AR markers position in the image is estimated. Once the error is less than a threshold, the UAV starts landing on the moving pad, minimizing the previous error.

In this work, we present a new algorithm for the tracking of an UGV by an UAV, using a video flow obtained from this UAV. This new vision-based tacking (VBT) method improves our previous works on the same subject [21], by incorporating a Fuzzy Logic Controller (FLC) in the process. Because most of drones parameters are unknown, the use of a Fuzzy Logic based approach is both relevant and suitable for reaching our purpose: autonomous tracking of an UGV by an UAV.

We use in our work the AR Drone 2.0 from Parrot [22]. The company provides the developers with a Software Development Kit (SDK) that allows them to control the UAV sending AT-commands and receiving data through Wifi communication protocol. In order to perform more complex tasks such as visual tracking or attitude control, the developer should use the most accurate physical model of the UAV, which is usually not provided by the company. Thus,

most people working on applications using this type of UAVs use a rough estimation of these parameters, which means rough results in tracking. For controlling such systems we think that a Fuzzy Logic Controller (FLC) is very suitable. Indeed, as said by Prof. Zadeh who introduced this concept in 1965, fuzzy logic is an "attempt to mimic human control logic" and this matches our goal of emulating the behavior of a professional pilot controlling manually a UAV over a moving UGV using visual feedback.

2. Problem statement

An UGV is guided by a human operator for inspecting a given area. In unknown environments, if the video flow from the UGV is the only source of information, there may be situations for which choosing a relevant path might be difficult. Adding an UAV that provides global coverage could greatly help the decision. Figure 1 shows both acquired images from the UGV and the UAV, we can notice that in the local coverage (the image on the right), the UGV has a limited vision on its surrounding, and in some cases next moves may be crucial. Thus adding the UAV video (left image), the operator has a clear view of the surrounding of the UGV, thus taking next move decision will be facilitated.

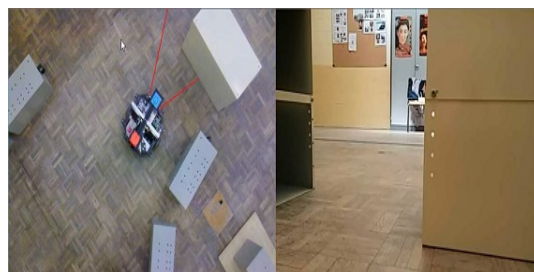


Figure 1. Global perception versus local perception

Continuously watching the video from the UAV, the pilot was able to track the UGV by keeping in real time this vehicle in the field of view of the camera. In figure 2 we can see the pilot (1) operating the drone (2), that takes a video of the navigation area and sends it to the ground station (3). The operator (4) selects new waypoints to guide the ground robot (5) between obstacles.



Figure 2. UAV pilot aided architecture

A second step was to render the tracking process of the UAV autonomous. For this purpose, we developed non linear controllers [21]. The developed controllers are model based. We used rough estimations of the UAVs parameters. The UAV was able to keep track of the UGV, but in order to improve our VBT system we propose in this work a FLC combined with a predictive algorithm, and show its overall efficiency compared to our previous implementations.

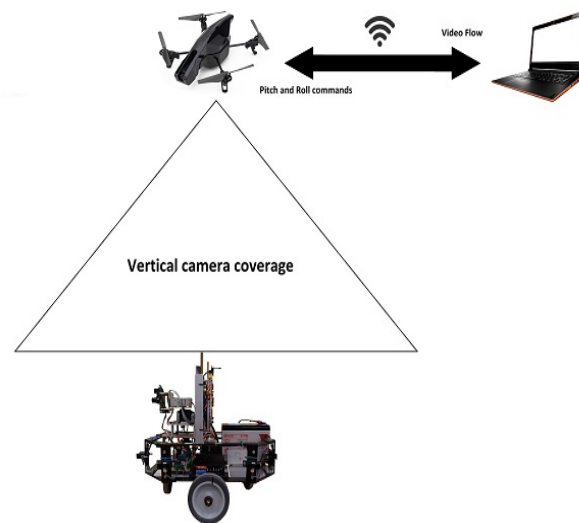


Figure 3. Autonomous tracking architecture

Figure 3 presents our autonomous tracking architecture. We can notice that it is a client/server based architecture, where the UAV flies over the tracked UGV, sends continuously the video flow acquired from the vertical camera to the ground station. The video flow is processed to extract the position of the tracked UGV in the image plan, and movement commands are sent back to the UAV in order to keep the UGV in the field of view of the UAV vertical camera.

It should be noted that the computation is performed remotely because of the limitations of the considered UAV (no payload). Apart from that point, nothing prevents the implementation of the processing on a capable UAV autopilot.

3. Vision based target tracking

We mentioned previously the advantages of using an AR marker over color detection. In previous works [21, 23] we installed an AR marker on the planar surface of the UGV. The size of the marker was chosen with respect to the low resolution of the UAV vertical camera. For marker identification, we use the library presented in [15]. Figure 4 represents the AR marker extracted from the library found in [24] that will be used later for the UGV's pose estimation.

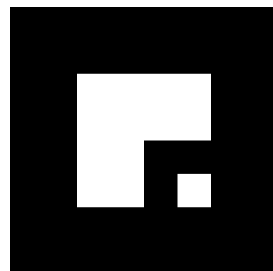


Figure 4. Tracked AR marker

We have to process each received image to check if the marker is present. Algorithm 1 illustrates the steps in order to estimate the 2D UGV's center (R_c) coordinates.

Algorithm 1: Augmented Reality (AR) marker tracking

Input: Image (video flow), AR marker (the tracked coded marker)

Output: X_{R_c}, Y_{R_c} (2D pose of the tracked marker)

```

begin
  Define AR marker (marker 1) ;
  for Image do
    if Image has marker 1 then
      Extract marker 1 corners (0, 1, 2, 3);
      Find the center of the marker 1;
      dx1 = x.corner(2)-x.corner(0);
      dy1 = y.corner(2)-y.corner(0);
      m1=dy1/dx1;
      c1=y.corner(0)-m1*x.corner(0);
      dx2 = x.corner(3)-x.corner(1);
      dy2 = y.corner(3)-y.corner(1);
      m2=dy2/dx2;
      c2=y.corner(1)-m2*x.corner(1);
       $X_{R_c} = (c2 - c1) / (m1 - m2)$ ;
       $Y_{R_c} = m1 * X_{R_c} + c1$ ;
      draw a red circle on ( $X_{R_c}, Y_{R_c}$ );
    else
      Continue searching;
  
```

Figure 5 shows the results of the AR marker (figure 4) tracking algorithm.

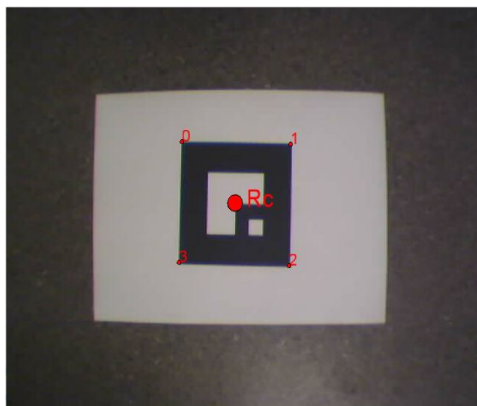


Figure 5. Augmented Reality (AR) marker tracking for UGV pose estimation

The main advantage of this method is that is more robust to lightning conditions compared to color tracking algorithm. Another advantage is that we can use different markers with different IDs in the case of multi-robot tracking. A natural environment most likely will not include an object with the same characteristics as an AR marker, which will limit the identification errors compared to color detection. The main drawback of this method is that at high altitudes, the marker will not be identifiable any longer, which will limit the flying height of the UAV, thus the covered area by the onboard vertical camera.

4. UGV pose estimation

We consider X_D and Y_D the axes of the image taken from the drone (Figure 6). In order to locate the UGV, we can use one of the presented tracking algorithms to extract the UGV's position (R_c). The position (in pixels) of R_c along X_D and Y_D axes is: Y_{Rc} and X_{Rc} . Y_0 and X_0 are the center (0, 0) of the image (we assume that the center of the image is the center of the drone).

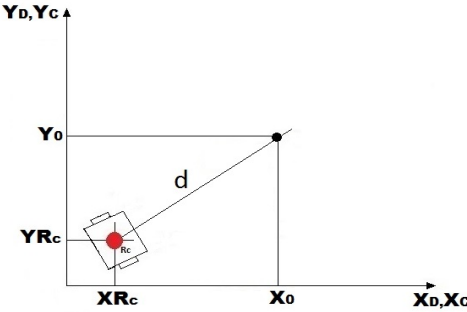


Figure 6. The UGV in the image plan

The distance d that separates the UGV and the center of the image can be written as follows:

$$d = \sqrt{(X_0 - X_{Rc})^2 + (Y_0 - Y_{Rc})^2} \tag{1}$$

Where X_{Rc}, Y_{Rc}, X_0, Y_0 represent the coordinates of the marker R_c and the center of the image. For the sequel, the three following hypotheses are supposed true:

Assumption 1. *The altitude of the UAV is supposed to be fixed during the UGV tracking.*

Assumption 2. *Due to their dynamic properties, the velocity of the UAV is higher compared to the one of the UGV.*

Assumption 3. *If the UGV is out of the field of view of the UAV, the latest stays in hover state.*

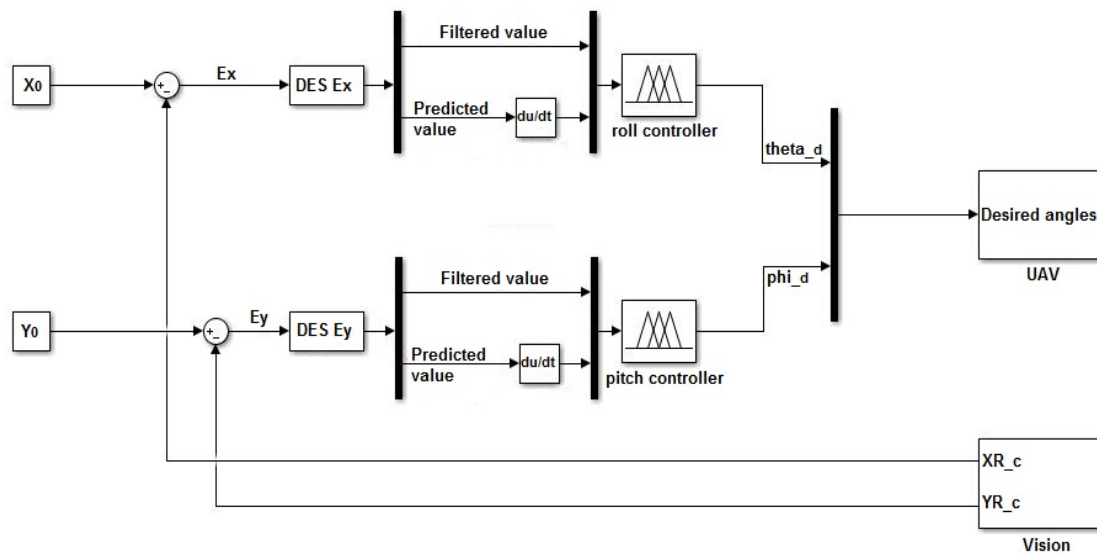


Figure 7. The structure of the autonomous tracking controller

5. Fuzzy Logic Control Design

Fuzzy Logic Controllers (FLC) have been widely used for vision based tracking. The tracked target related location is extracted by estimating the pose of a unique visual feature that belongs to the target, which should be inexistent in the surrounding area. A simple way to do so is to track a color. A red balloon is used in [25] to simulate a flying object. The distance and the orientation to the target are estimated using the size and the position of the balloon in the image plan. The outputs of the FLC to track the flying object are pitch and yaw commands. Artificial markers (AR markers for example) can also be used to estimate the pose of the tracked target. They are usually placed on a planar surface on the target, and the inputs to the FLC are the X and Y errors between the center of the image and the center of the marker. Vision based FLC is used in other works for navigation [26, 27], and autonomous landing [28, 29].

As Prof. Zadeh introduced in 1965, fuzzy logic is an *"attempt to mimic human control logic"*. We developed in this work a FLC to imitate a professional pilot, mainly to give our system more autonomy. A first step in creating a FLC to control the UAV consists in creating fuzzy sets. A fuzzy set is the entity that represents an input or an output of a system. In our case, the input is the pose estimation of the UGV in the image plan, and the output is the desired roll and pitch angles in order to follow autonomously the UGV. Taking the example of an input, the fuzziness of the set is illustrated in the degree of belonging to the different states of this set. In the UAV image plan, the UGV can be far negative (N), at the center (Z), or far positive (P) distance from the center of the image. These are called membership functions, where we define the degree of belonging of the crisp values of the UGV pose to these states (N, Z, P). Once we have our input membership functions, the same process is applied to extract the output, and the bridge between the two values (input and output) is what is called Defuzzification. We will see in next section more details on the different membership functions of the FLC and their defuzzification in order to get the desired attitude of the UAV for the autonomous target tracking.

As the UGV is supposed to move on a planar surface, the two outputs that needs to be fuzzified in order to follow autonomously the UGV are roll and pitch angles, which are respectively the angles necessary to change in order to move along X and Y axes of the UAV. The input to the FLC is then location of the UGV in the image plan, which is in our work the AR marker X and Y position of its center obtained as explained in previous section.

Our data processing is based on a client/server architecture. Captured video flow is sent continuously to a ground station where it will be processed. Once the targets location extracted, the pose errors to the center of the image are estimated. Double Exponential Smoothing (DES) algorithm is used to filter the measurements, and to give a predicted value of the pose estimation. The output of the DES algorithm are fed to the FLC in order to generate the necessary output to track the target. The generated roll and pitch commands are then sent through the chosen communication protocol to the UAV.

We explain in this section the fuzzification of the different inputs and outputs of the controller, the used method for defuzzification, as well as the DES algorithm.

5.1 Inputs and outputs fuzzification

To enable the autonomous tracking of the UGV, the acquired video flow from the vertical camera of the UAV is processed. The results, as explained in algorithm 1, are the X and Y coordinates of the marker in the image plan. Since we suppose that the optical center of the camera and the center of the UAV are the same, the extracted coordinates are the relative position of the UGV to the UAV in a planar surface. The errors to be considered then are the ones between the center of the UAV X_0, Y_0 , and the UGV coordinates X_{Rc}, Y_{Rc} :

$$e_{X_D} = X_{Rc} - X_0 \quad e_{Y_D} = Y_{Rc} - Y_0 \quad (2)$$

The control objective can be expressed as follows:

$$\lim_{t \rightarrow \infty} e_{X_D}(t) = 0 \quad \lim_{t \rightarrow \infty} e_{Y_D}(t) = 0 \tag{3}$$

To fulfill this objective, the UAV has to move along its X and Y axes, the outputs of the FLC are in this case the roll ϕ_d and pitch θ_d angles to be fed to the UAV autopilot.

Two FLCs were designed to achieve the control objective (3). A representation of the proposed autonomous tracking controller is shown in figure 7. We used a quadcopter UAV, and we suppose that it has a symmetrical construction, this means that inertia moments along X_D and Y_D are identical. In this case pitch and roll controllers are identical. Thus for brevity, we will present the membership functions of the inputs and output for one of them, the other one is considered to be exactly the same.

To deal with the uncertainties of the estimated pose of the AR marker, the Double Smoothing Exponential (DES) algorithm is implemented. The DES algorithm can also provide us with the predicted value of the errors. The output of the DES are then the filtered values of the errors along X or Y axes, and their predicted value in a predefined horizon. We will present in next section the DES algorithm and the implementation of the proposed FLC.

5.2 Double Exponential Smoothing Algorithm

It has been show in [30, 31] that the DES algorithm runs approximately 135 times faster with equivalent prediction performances and simpler implementations compared to a Kalman filter. It has been successfully implemented in [32] for a vision based target tracking, and showed good results for colored blob tracking with the same UGV used in our present experiments.

The DES algorithm is implemented for the projection of the center of the AR marker X_{Rc} and Y_{Rc} separately as shown in figure 7. It is given as follows:

$$S_{m_n} = \alpha_m \cdot m_n + (1 - \alpha_m) \cdot (S_{m_{n-1}} + b_{m_{n-1}}) \tag{4}$$

$$b_{m_n} = \lambda_m \cdot (S_{m_n} - S_{m_{n-1}}) + (1 - \lambda_m) \cdot b_{m_{n-1}} \tag{5}$$

Where:

m_n is the value of (X_{Rc} or Y_{Rc}) at n th sample instant.

S_{m_n} is the smoothed value of (X_{Rc} or Y_{Rc}).

b_{m_n} is the trend value of (X_{Rc} or Y_{Rc}).

α_m is the smoothing factor.

λ_m is the trend smoothing factor.

Equation (4) smooths the value of the sequence of measurements by taking into account the trend, whilst (5) smooths and updates the trend.

We denote $k_f > 0$ the number of steps in the future for the position of the AR marker (X_{Rc} , Y_{Rc}). The forecasted value can be written as follows:

$$F_{m_{n+k_f}} = S_{m_n} + k_f \cdot b_{m_n} \tag{6}$$

The initial values given to S_{m_n} , b_{m_n} are:

$$S_{m_1} = m_1 \quad b_{m_1} = m_2 - m_1 \quad (7)$$

A compromise has to be found for the values of α_m ($0 \leq \alpha_m \leq 1$) and λ_m ($0 \leq \lambda_m \leq 1$). 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 FLC.

5.3 Membership functions

Roll controller has two inputs, the filtered error and the derivative of its predictive value. Both the inputs and the output of the controller have triangular membership functions, their selection was based on their performance to achieve the desired objective of the controller (3).

Figures 8, 9, 10, show respectively the membership functions of the filtered value e_{XD_f} , the derivative of the predicted value e_{XD_p} , and the desired roll angle θ_d .

The fuzzy sets of the inputs e_{XD_f} and e_{XD_p} are:

- (1) N: Negative
- (2) Z: Zero
- (3) P: Positive

The fuzzy sets of the output θ_d are:

- (1) NB: Negative Big
- (2) N: Negative
- (3) Z: Zero
- (4) P: Positive
- (5) PB: Positive Big

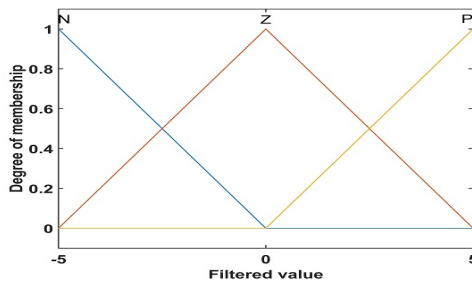


Figure 8. Membership function of the filtered error

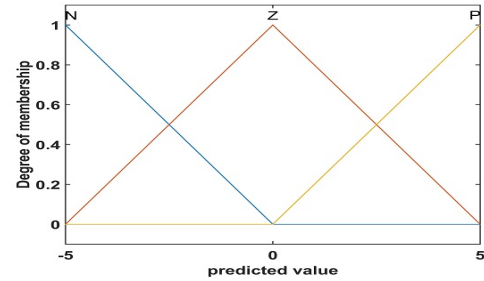


Figure 9. Membership function of the derivative of the predicted error

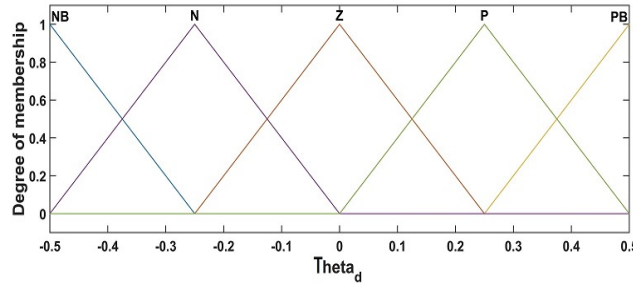


Figure 10. Membership function of the desired roll angle

5.4 Defuzzification

We have used Mamdani min-max for the Fuzzy Inference System (FIS). The defuzzification method is centroid based. The FIS has nine rules. Figure 11 shows the FIS inputs and output surface. More details on the impact of the form and number of fuzzy sets have been discussed in [33, 34].

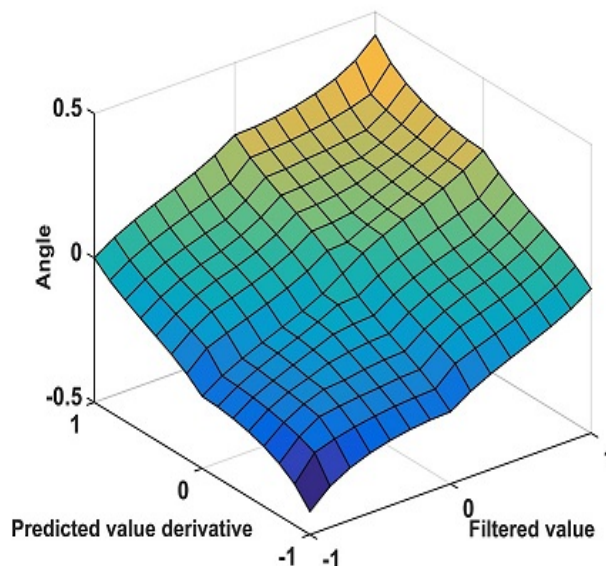


Figure 11. Fuzzy membership surface for the inputs and the output of the controller

Inference of inputs results from the use of "min" operator (for conjunction and implication) and "max" operator (for disjunction and aggregation). Fuzzy rules are summed up in Table 1.

e_{XD_p}	N	Z	P
e_{XD_f}	NB	N	Z
N	NB	N	Z
Z	N	Z	P
P	Z	P	PB

Table 1. FLC rules

6. Simulation results

Matlab Simulink has been used to simulate the proposed FLC controller with the DES algorithm. We have used in this work the simplified dynamic model of a quadcopter presented in [35]. For generating the FIS, we used the fuzzy logic toolbox in Matlab Simulink. The toolbox simplifies the coding tasks since it can be used easily to create the inputs, outputs, and the defuzzification methods of the system.

For the autonomous tracking, the UGV model has been presented in [32]. The UGV follows a sinusoidal trajectory, and the UAV, located initially at (4,0) in the world frame takes off, hovers at 3 meters of altitude, then starts the autonomous tracking of the UGV.

We have used for simulation the following parameters:
Simulation time: 200s.

Sampling period: $T_e = 0.01s$.
 DES algorithm: $\alpha_m = 0.7$, $\lambda_m = 0.3$, $k_f = 2$.

Figure 12 presents the trajectory of both UGV and the UAV.

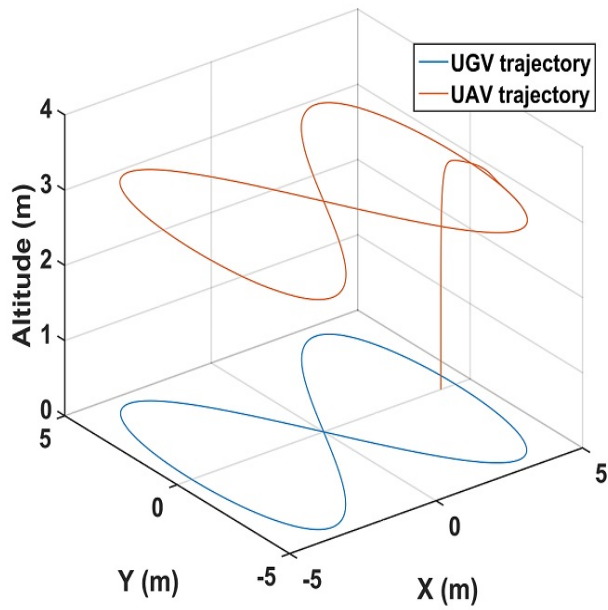


Figure 12. Autonomous tracking of an UGV

The angles variations of the UAV are directly related to the errors along X and Y axes between the center of the UGV and the center of the image of the UAV, presumed previously to be the center of the UAV. Figures 13 and 14 show the variation of roll and pitch angles respectively.

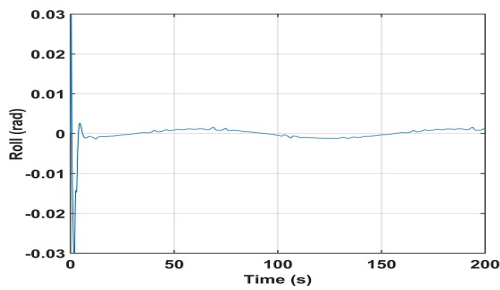


Figure 13. Roll angle variation

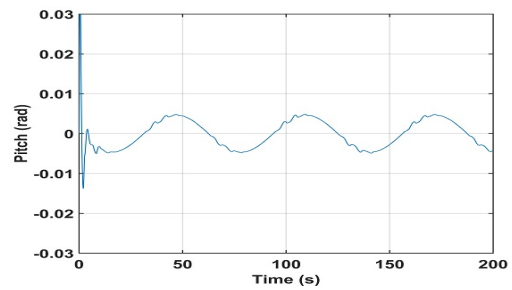


Figure 14. Pitch angle variation

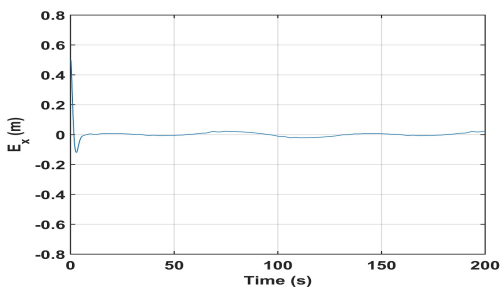


Figure 15. The tracking errors along X axis

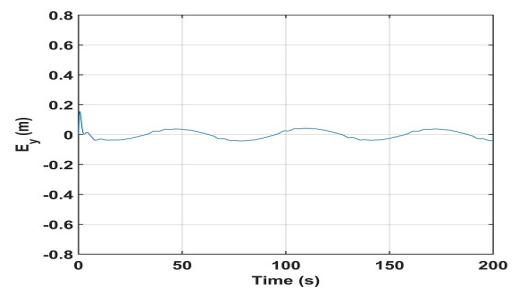


Figure 16. The tracking errors along Y axis

The evolution of the tracking errors along X and Y axis between the UAV and UGV positions are shown in figure 15 and 16 respectively. Figures 12, 15, and 16 show that the control objective (3) has been achieved.

6.1 DES parameters tuning

As we explained in section 5.2, a compromise has to be found between the values of α_m and λ_m respectively. We have conducted several simulations to obtain the best combination of these values. Figures 17 and 18 illustrate the results of the errors evolution along X and Y axis respectively using a fixed value of $\lambda_m = 0.1$, and variable value of $\alpha_m = [0.1, 0.3, 0.6, 0.9]$. While in figures 19 and 20 the results of the simulation are obtained using a fixed value of $\alpha_m = 0.1$, and variable value of $\lambda_m = [0.3, 0.6, 0.9]$.

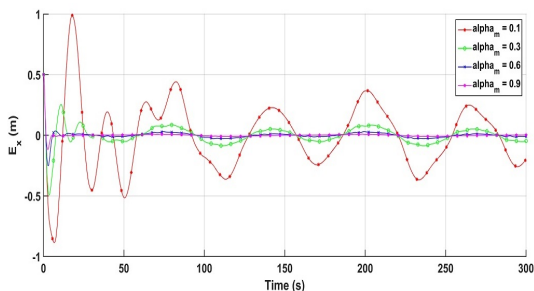


Figure 17. α_m variation influence on E_x

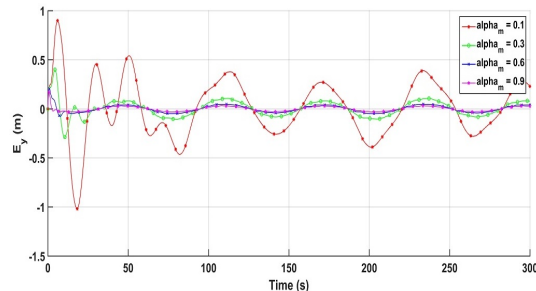


Figure 18. α_m variation influence on E_y

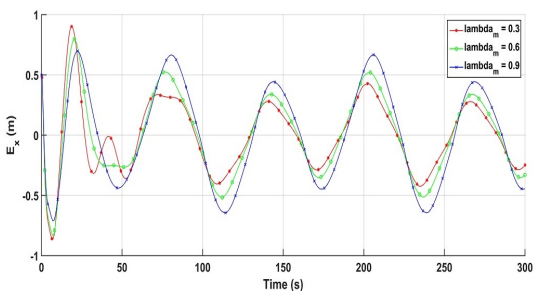


Figure 19. λ_m variation influence on E_x

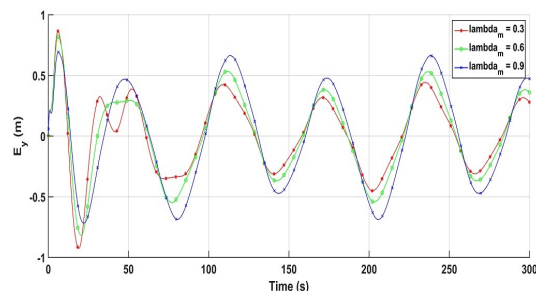


Figure 20. λ_m variation influence on E_y

Simulations have been conducted initially with the horizon $k_f = 2$. We will see now the influence of the horizon k_f changing with the defined values of α_m , and λ_m .

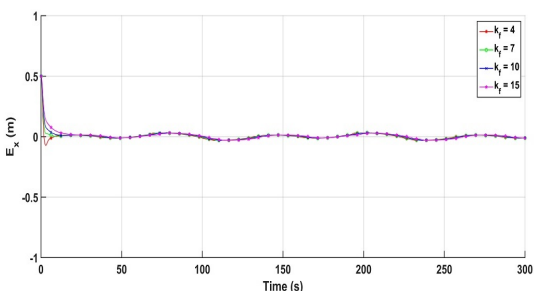


Figure 21. k_f variation influence on E_x

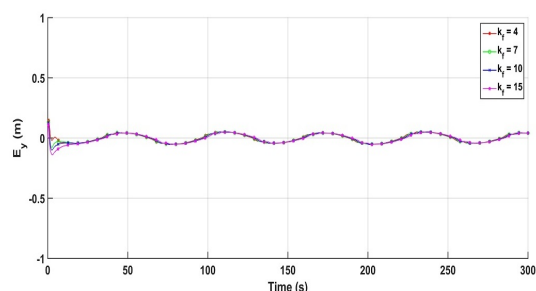


Figure 22. k_f variation influence on E_y

We can see in figures 21 and 22 the different values of the errors E_x and E_y according to the variation of the step k_f . We notice that in simulation the changing of the value of k_f does not influence greatly the autonomous tracking, mainly because the measurement of the UGV pose

is provided directly by the software, thus it does not have the uncertainties in pose estimation similar to real experiments.

Simulations have been carried out in the world frame, where the access to the UAV and the UGV positions are easily extracted. In real experimentations, the tracking will be in the image frame, thus the inputs will have a different scale (from meters to pixels), and the output from radians to a percentage of the maximum tilt of the used UAV. We will explain in next section the different steps to implement the proposed controller in real experiments, and discuss its efficiency compared with our previous work [21].

7. Experimental results

Experiments have been carried out using an AR Drone 2.0 of Parrot [22]. The video flow of the UAV is sent continuously to the ground station, to be processed in our developed Graphical User Interface (GUI), then send back roll and pitch values to the UAV through a WIFI link.

The tracked target is a non-holonomic Wheeled Mobile Robot (WMR), developed in our laboratory. To extract the position of the UGV, we printed the AR marker (figure 4), and placed it upon the UGV on a planar surface. The UGV is moved manually in a random way by a human operator through the same GUI. The commands are sent to the UGV through a ZigBee communication protocol.

We implemented for the experimentation the FLC in a Java based environment called Processing [36]. It is an open source coding language used initially for art projects, which is gaining popularity for its simplistic design for learning and prototyping. The main advantage of using Processing is the easiness of implementation when it comes to include many libraries at the same time, for instance, AR marker detection, WIFI and ZigBee communication, graphical materials, and so on.

To implement the FLC in Processing, we used a Fuzzy Associative Matrix (FAM) representation [37] for the input, output, and defuzzification of our membership functions. Its an easier way than to use IF-THEN list, since it presents the system in a tabular way firing the output of the system according to the given inputs. The main drawback of the FAM is its complexity with the implementation if the system has more than two inputs, but since we have only two inputs (the filtered value and the derivative of the predictive value), FAM suits our purpose very well.

The tracking in real experiments is done in the image frame, the only difference compared to simulations is in the scale of the input values. It was expressed in the world frame from $[-1, 1]m$ for both inputs, to be changed in the experimentations for the following values (only the errors along X axis will be considered, the ones along Y axis are identical): Filtered values: $[-224, +224]$ pixels. The derivative of predicted values: $[-320, +320]$ pixels.

The output scale varies from $[-maxtilt_{r/p}, +maxtilt_{r/p}]$ with the maximum tilt roll and pitch values ($maxtilt_{r/p}$) are to be defined experimentally, knowing that a big value may render the system instable with the risk of losing the target from the field of view.

The parameters are tuned experimentally to get the best performance. The difference between the parameters used in simulation and those used in experimentations are simply due to the lack of the exact UAV model, thus the difference between the simulated UAV and the real one. The following parameters have been used in the experiments:

$maxtilt_r = 0.15$, $maxtilt_p = 0.1$.

Sampling period: $T_e = 0.01s$.

DES algorithm: $\alpha_m = 0.8$, $\lambda_m = 0.1$, $k_f = 1$.

The images are sent continuously from the AR Drone 2.0 via a WIFI link to the ground station which is a laptop with an intel i5 processor running at 1.7GHz and 4Gb of ram. Each image is then processed to extract the AR maker position, and to solve the FLC outputs. The commands are then sent to the AR Drone 2.0 in order to track the UGV. The exact processing time is not available because it may be affected by the background programs running at the ground station

Operating System (OS). Nevertheless, a rough estimation of the control loop of the developed code running on the aforementioned laptop showed that the processing time is at approximately 50ms for each loop.

Figure 23 represents the GUI developed with Processing. We can see that the tracked marker has been identified in the image, and the estimated errors between its center and the center of the image is fed to the FLC. The input memberships are drawn on the bottom of the GUI (to the left are the inputs of the roll controller, and to the right are the inputs of the pitch controller).



Figure 23. The Graphical User Interface (GUI) overview

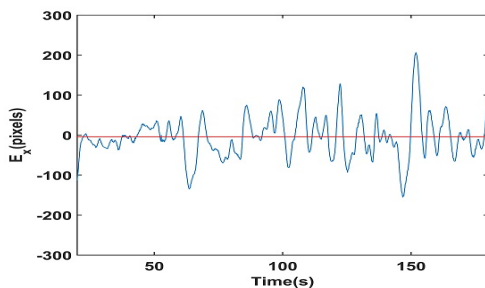


Figure 24. Experimental tracking errors along X axis

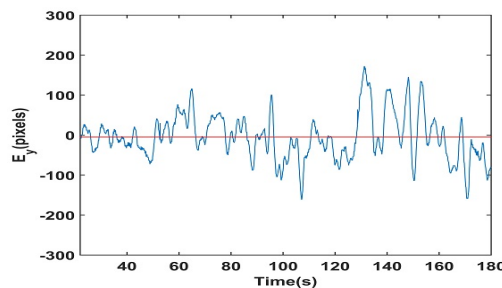


Figure 25. Experimental tracking errors along Y axis

In real world scenarii, for the case of an inspection or exploration mission, the UGV follows a random trajectory. To meet this requirement in our experiments, the human operator drives manually the UGV by selecting random waypoints in the image plan [38]. The trajectories of both UAV and UGV has not been measured because of the lack of an exteroceptive system (Vicon for example), and the internal sensors (IMU, odometry) provides poor results due to drift in their measurements. Nevertheless, the tracking errors along X and Y axis between the UAV and the UGV are measured in the image plan of the UAV. They are shown in figures 24 and 25 respectively. We can see from these figures that the mean values of E_x and E_y are around zero, which validates the tracking objective (3), and this means that the UAV is able to autonomously follow the UGV in its random waypoints tracking.

8. Performance discussion

We discuss in this section the results of the visual tracking using firstly the Proportional Derivative (PD) controller, then the FLC controller with inputs the error and its derivative, and finally the FLC coupled with DES algorithm. In the experiments, the same presumptions are used: random trajectory of the UGV with the same velocity, same altitude (with +/- 20 cm). The tracking begins when the UGV is in the field of view of the UAV.

The tracking error comparison is based on the distance d (1) between the center of the image (supposed to be the center of the UAV), and the center of the AR marker (supposed to be the center of the tracked UGV).

8.1 Proportional Derivative (PD) controller

We presented in [21] a PD controller for vision based tracking. Equation 8 illustrates the desired angles to be sent to the UAV in order to track the UGV:

$$\begin{aligned} \phi_d &= \text{asin}(-u_{Y_D}) & \text{where} & & u_{Y_D} &= \frac{m}{U_1}((K_1 e_{Y_D} + K_2 \dot{e}_{yD}) \\ \theta_d &= \text{asin}\left(\frac{u_{X_D}}{\cos\phi_d}\right) & \text{where} & & u_{X_D} &= \frac{m}{U_1}(K_1 e_{X_D} + K_2 \dot{e}_{x_D}) \end{aligned} \quad (8)$$

K_1 and K_2 are positive gains. U_1 is called the total thrust (U_1), and m is the UAVs weight. In simulation the value of the total thrust U_1 can be obtained easily based on the used UAV model. In the case of a quadrotor type UAV, U_1 is the sum of the four rotors forces, that is based on the thrust factor and the rotation speed of the propellers. In real experiments, when using the AR Drone 2.0, the value of U_1 is roughly estimated, since the exact model of the UAV is not available, and may be cumbersome to obtain using estimations for each part, resulting in rough tracking of the UGV. The experimental results using PD controller is shown in figure 26. The tracking error has a mean value of around 124 pixels.

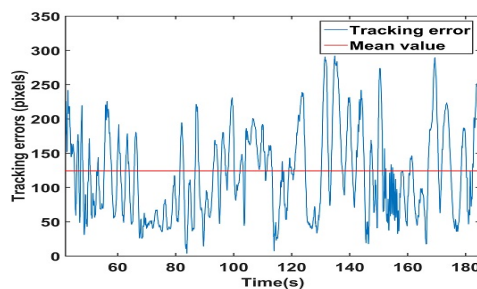


Figure 26. Tracking error using Proportional Derivative Controller [21]

8.2 Fuzzy Logic Controller without DES algorithm

In the FLC we have used the same membership functions described in this paper, with the difference of the inputs of the controller and their scales.

The tracking errors of the FLC (figure 27) were promising compared to the previous PD controller. The mean value of the tracking errors is around 96 pixels.

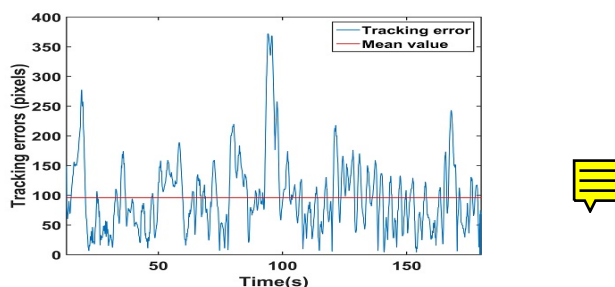


Figure 27.Tracking error using FLC without DES

8.3 Fuzzy Logic Controller with DES algorithm

DES algorithm has been used to filter the uncertainties of the estimated pose of the AR marker, supposed to represent the tracked UGV. We presented in section 5.2 the algorithm, and used its outputs (filtered and predicted values) as an input to the FLC instead of direct measured values. Figure 7 illustrates the FLC with the DES algorithm for our vision based target tracking. The experimental results of the tracking are shown in figure 28.

The tracking errors using the FLC with the DES algorithm gave a mean value of around 65 pixels. A video of the experimental results can be found online [39].

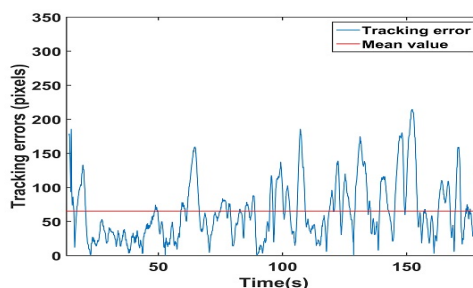


Figure 28.Distance error using the FLC with DES algorithm

8.4 Comparison

The tracking error expressed in pixels that separates the center of the UAV and the center of the UGV in the plan image is used as a reference for the experimental results comparison. As we explained previously, each experiment is conducted in the same manner, with a randomly driven UGV with the same velocity, and same flying altitude of the UAV. The improvement of the autonomous vision based target tracking is obtained using the FLC instead of the PD controller, and it has been further improved when coupling the FLC with the DES algorithm, resulting in a smoother and more accurate tracking. Table 2 summarize the experimental results obtained with each controller.

	PD	FLC	FLC with DES
Min	3.726	0	0
Max	292.1	372	215
Mean	124.3	96.12	65.19
Improvement	-	22.67 %	47.55 %

Table 2.Performance analysis of the different controllers

The obtained results states clearly the improvement of the autonomous vision based tracking using the FLC with DES algorithm. The mean value of the tracking errors obtained using this controller reduced with 47.55 % the tracking errors compared to the PD controller, against the 22.67 % using the FLC.

9. Conclusion and Perspectives

We have presented in this paper a Fuzzy Logic Controller for an UGV tracking using an UAV. Unlike conventional controllers, FLC does not require a prior knowledge of the system modeling. This is particularly interesting when using commercial UAVs, where the detailed parameters of the UAVs model are unknown, and sometimes difficult to be estimated.

Since we lack model information of our experimental platform, FLC can be considered as a good choice for autonomous tracking. The fuzzification of the inputs and outputs of the FLC are based on the feedback of an experienced pilot. The overall system is a client/server architecture. The captured video from the UAV is sent continuously to a ground station, where we developed a GUI to visualize the video flow, and to process the data in order to extract the position of the tracked target. FLC was implemented using a Fuzzy Associative Matrix (FAM) in order to send the necessary pitch and roll commands back to the UAV through WIFI network.

We illustrated in this work the tracking scheme. The acquired measurements have been filtered using a DES algorithm. The same algorithm generates the predictive values of the tracking errors, that will be in addition to the filtered values considered as the input of our FLC. The proposed FLC was validated through simulations and experimentations. It has been shown that this work reduced by 47.55 % the tracking errors compared to our previous work [21] where we used a Proportional Derivative controller. A future work consists in implementing the proposed FLC controller directly inside an UAV autopilot, mainly to deal with the problem of communication delays related to the client/server architecture nature.

Acknowledgment

The authors would like to thank Le Havre Town Council CODAH for their support under research grants. They would like as well to thank Haute-Normandie Region (France) through their support to PCMAI project.

References

- [1] Anthony Stentz, Alonzo Kelly, Herman Herman, Peter Rander, Omead Amidi, and Robert Mandelbaum. Integrated air/ground vehicle system for semi-autonomous off-road navigation. *Robotics Institute*, page 18, 2002.
- [2] Nicolas Vandapel, Raghavendra Donamukkala, and Martial Hebert. Experimental results in using aerial ladar data for mobile robot navigation. In *Field and Service Robotics*, pages 103–112. Springer, 2006.
- [3] Lars Kuhnert., Klaus Müller., Markus Ax., and Klaus-Dieter Kuhnert. "object localization on agricultural areas using an autonomous team of cooperation ground and air robots". In *International Conference of Agricultural Engineering, CIGR-AGENG2012, Valencia, Spain, July 8-12 2012*.
- [4] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Air-ground localization and map augmentation using monocular dense reconstruction. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3971–3978. IEEE, 2013.
- [5] Nathan Michael, Shaojie Shen, Kartik Mohta, Yash Mulgaonkar, Vijay Kumar, Keiji Nagatani, Yoshito Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, et al. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841, 2012.

- [6] Martin Saska, Tomáš Krajník, and Libor Pfeucil. Cooperative μ uav-ugv autonomous indoor surveillance. In *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, pages 1–6. IEEE, 2012.
- [7] Francesco Cocchioni, Valerio Pierfelice, Alessandro Benini, Antonella Mancini, Emanuele Frontoni, Primo Zingaretti, Gianluca Ippoliti, and Sauro Longhi. Unmanned ground and aerial vehicles in extended range indoor and outdoor missions. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, pages 374–382. IEEE, 2014.
- [8] Cheng Hui, Chen Yousheng, Li Xiaokun, and Wong Wing Shing. Autonomous takeoff, tracking and landing of a uav on a moving ugv using onboard monocular vision. In *Control Conference (CCC), 2013 32nd Chinese*, pages 5895–5901. IEEE, 2013.
- [9] Jae-Keun Lee, Hahmin Jung, Huosheng Hu, and Dong Hun Kim. Collaborative control of uav/ugv. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2014 11th International Conference on*, pages 641–645. IEEE, 2014.
- [10] Matthias Faessler, Elias Mueggler, Karl Schwabe, and Davide Scaramuzza. A monocular pose estimation system based on infrared leds. *International Conference on Robotics and Automation (ICRA)*, 2014.
- [11] Jae-Young Choi and Sung-Gaun Kim. *Collaborative Tracking Control of UAV-UGV*. *World Academy of Science, Engineering and Technology*, 71, 2012.
- [12] René Vidal, Omid Shakernia, H Jin Kim, David Hyunchul Shim, and Shankar Sastry. Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *Robotics and Automation, IEEE Transactions on*, 18(5):662–669, 2002.
- [13] H Jin Kim, Rene Vidal, David H Shim, Omid Shakernia, and Shankar Sastry. A hierarchical approach to probabilistic pursuit-evasion games with unmanned ground and aerial vehicles. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 1, pages 634–639. IEEE, 2001.
- [14] Luiz Chaimowicz, Ben Grocholsky, James F Keller, Vijay Kumar, and Camillo J Taylor. "Experiments in multirobot air-ground coordination". In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 4053–4058. IEEE, 2004.
- [15] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85–94. IEEE, 1999.
- [16] Mark Fiala. Vision guided control of multiple robots. In *Computer and Robot Vision, 2004. Proceedings. First Canadian Conference on*, pages 241–246. IEEE, 2004.
- [17] Tomáš Krajník, Matías Nitsche, Jan Faigl, Petr Vaněk, Martin Saska, Libor Přeučil, Tom Duckett, and Marta Mejail. A practical multirobot localization system. *Journal of Intelligent & Robotic Systems*, 76(3-4):539–562, 2014.
- [18] ArUco: a minimal library for augmented reality applications based on opencv., 2015. URL: <http://www.uco.es/investiga/grupos/ava/node/26>.
- [19] Matevž Bošnjak, Drago Matko, and Sašo Blažič. Quadcopter hovering using position-estimation information from inertial sensors and a high-delay video system. *Journal of Intelligent & Robotic Systems*, 67(1):43–60, 2012.
- [20] Daewon Lee, Tyler Ryan, and H Jin Kim. Autonomous landing of a vtol uav on a moving platform using image-based visual servoing. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 971–976. IEEE, 2012.
- [21] Harik El Houssein Chouaib, Guérin François, Guinand Frédéric, Brethé Jean-François, and Pelvillain Hervé. A decentralized interactive architecture for aerial and ground mobile robots cooperation. In *International Conference on Control, Automation and Robotics (ICCAR)*, pages 37–43. IEEE, 2015.
- [22] Ar Drone 2.0, 2015. URL: <http://ardrone2.parrot.com/>.
- [23] Harik El Houssein Chouaib, Guérin François, Guinand Frédéric, Brethé Jean-François, Pelvillain Hervé, and Zentout Adel. Vision based target tracking using an unmanned aerial vehicle. In *IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO 2015)*, page 6, Lyon, France. 2015. URL: <https://hal-insu.archives-ouvertes.fr/UNIV-LEHAVRE/hal-01170117v1>.
- [24] ARToolkit Patternmaker, 2015. URL: <http://www.cs.utah.edu/gdc/projects/augmentedreality/>.
- [25] Miguel A Olivares-Mendez, Ivan Mondragon, Pascual Campoy Cervera, Luis Mejias, and Carol Martinez. Aerial object following using visual fuzzy servoing. In *Proceedings of the 1st Workshop on Research, Development and Education on Unmanned Aerial Systems (RED-UAS 2011)*, pages 61–70. Centro Avanzado de Tecnologías Aeroespaciales (CATEC), 2011.
- [26] Gianluca Antonelli, Stefano Chiaverini, and Giuseppe Fusco. A fuzzy-logic-based approach for mobile

- robot path tracking. *IEEE Transactions on Fuzzy Systems*, 15(2):211–221, 2007.
- [27] Patrick Benavidez and Mo Jamshidi. Mobile robot navigation and target tracking system. In *System of Systems Engineering (SoSE), 2011 6th International Conference on*, pages 299–304. IEEE, 2011.
- [28] Andrea Cesetti, Emanuele Frontoni, Adriano Mancini, Primo Zingaretti, and Sauro Longhi. A vision-based guidance system for uav navigation and safe landing using natural landmarks. In *Selected papers from the 2nd International Symposium on UAVs, Reno, Nevada, USA June 8–10, 2009*, pages 233–257. Springer, 2009.
- [29] Jeffrey W Tweedale. Fuzzy control loop in an autonomous landing system for unmanned air vehicles. In *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*, pages 1–8. IEEE, 2012.
- [30] Joseph J LaViola Jr. "an experiment comparing double exponential smoothing and kalman filter-based predictive tracking algorithms". In *Virtual Reality, 2003. Proceedings. IEEE*, pages 283–284. IEEE, 2003.
- [31] Joseph J LaViola. Double exponential smoothing: an alternative to kalman filter-based predictive tracking. In *Proceedings of the workshop on Virtual environments 2003*, pages 199–206. ACM, 2003.
- [32] François Guerin, Simon G. Fabri, and Marvin K. Bugeja. "double exponential smoothing for predictive vision based target tracking of a wheeled mobile robot". In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 3535–3540, Florence, Italy, December 2013.
- [33] John H Lilly. *Fuzzy control and identification*. John Wiley & Sons, 2011.
- [34] Hao Ying. *Fuzzy control and modeling: analytical foundations and applications*. Wiley-IEEE Press, 2000.
- [35] Samir Bouabdallah and Roland Siegwart. Full control of a quadrotor. In *Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on*, pages 153–158. IEEE, 2007.
- [36] Processing, 2015. URL: <https://processing.org/>.
- [37] Fuzzy Logic Control using Processing, 2016. URL: http://www.xfirm.com/processing/Fuzzy_logic/.
- [38] Harik El Houssein Chouaib, Guérin François, Guinand Frédéric, Brethé Jean-François, and Pelvillain Hervé. Uav-ugv cooperation for objects transportation in an industrial area. In *IEEE International Conference on Industrial Technology (ICIT)*, pages 547–552. IEEE, 2015.
- [39] Experiment video of the FLC., 2016. URL: <https://youtu.be/9yfo5McsA6U>.