




Article

# Monocular Visual SLAM Based on a Cooperative UAV–Target System

Juan-Carlos Trujillo <sup>1</sup>, Rodrigo Munguia <sup>1,\*</sup>, Sarquis Urzua <sup>1</sup>, Edmundo Guerra <sup>2</sup> and Antoni Grau <sup>2</sup>

<sup>1</sup> Department of Computer Science, CUCEI, University of Guadalajara, Guadalajara 44430, Mexico; juancarlos\_max@hotmail.com (J.-C.T.); isi.sarquis@gmail.com (S.U.)

<sup>2</sup> Department of Automatic Control, Technical University of Catalonia UPC, 08034 Barcelona, Spain; edmundo.guerra@upc.edu (E.G.); antoni.grau@upc.edu (A.G.)

\* Correspondence: rodrigo.munguia@academicos.udg.mx

Received: 23 March 2020; Accepted: 18 June 2020; Published: 22 June 2020



**Abstract:** To obtain autonomy in applications that involve Unmanned Aerial Vehicles (UAVs), the capacity of self-location and perception of the operational environment is a fundamental requirement. To this effect, GPS represents the typical solution for determining the position of a UAV operating in outdoor and open environments. On the other hand, GPS cannot be a reliable solution for a different kind of environments like cluttered and indoor ones. In this scenario, a good alternative is represented by the monocular SLAM (Simultaneous Localization and Mapping) methods. A monocular SLAM system allows a UAV to operate in a priori unknown environment using an onboard camera to simultaneously build a map of its surroundings while at the same time locates itself respect to this map. So, given the problem of an aerial robot that must follow a free-moving cooperative target in a GPS denied environment, this work presents a monocular-based SLAM approach for cooperative UAV–Target systems that addresses the state estimation problem of (i) the UAV position and velocity, (ii) the target position and velocity, (iii) the landmarks positions (map). The proposed monocular SLAM system incorporates altitude measurements obtained from an altimeter. In this case, an observability analysis is carried out to show that the observability properties of the system are improved by incorporating altitude measurements. Furthermore, a novel technique to estimate the approximate depth of the new visual landmarks is proposed, which takes advantage of the cooperative target. Additionally, a control system is proposed for maintaining a stable flight formation of the UAV with respect to the target. In this case, the stability of control laws is proved using the Lyapunov theory. The experimental results obtained from real data as well as the results obtained from computer simulations show that the proposed scheme can provide good performance.

**Keywords:** state estimation; unmanned aerial vehicle; monocular SLAM; observability; cooperative target; flight formation control

---

## 1. Introduction

Nowadays, unmanned aerial vehicles (UAVs), computer vision techniques, and flight control systems have received great attention from the research community in robotics. This interest has resulted in the development of systems with a high degree of autonomy. UAVs are very versatile platforms and very useful for several tasks and applications [1,2]. In this context, a fundamental problem to solve is the estimation of the positions of UAVs. For most applications, GPS (Global Positioning System) still represents the main alternative for addressing the localization problem of UAVs. However, GPS comes with some well-known drawbacks associated with its use. For instance, in scenarios where GPS signals are jammed intentionally [3] or when the precision error can be

substantial and they provide poor operability due to multipath propagation (e.g., natural and urban canyons [4,5]). Furthermore, there are scenarios where the GPS is inaccessible (e.g., indoor). Hence, to improve accuracy and robustness, additional sensory information, like visual data, can be integrated into the system. Cameras are lightweight, inexpensive, power-saving, and provide lots of information, moreover, they are well adapted to be integrated into embedded systems. In this context, visual SLAM methods are important options that allow a UAV to operate in an a priori unknown environment using only on-board sensors to simultaneously build a map of its surroundings while, at the same time, locating itself in respect to this map. On the other hand, perhaps the most important challenge associated with the application of monocular SLAM techniques has to do with the metric scale [6]. In monocular SLAM systems, the metric scale of the scene is difficult to retrieve, and even if the metric scale is known as an initial condition, the metric scale tends to degenerate (drift) if the system does not incorporate continuous metric information.

Many works can be found in the literature where visual-based SLAM methods are used for UAV navigation tasks (e.g., [7,8]). For SLAM based on monocular vision, different approaches have been followed for addressing the problem of the metric scale. In [9], the position of the first map features is determined by knowing the metrics of an initial pattern. In [10], a method with several assumptions about the structure of the environment is proposed; one of these assumptions is the flatness of the floor. This restricts the use of the method to very specific environments. Other methods like [11] or [12] fuse inertial measurements obtained from an inertial measurement unit (IMU) to recover the metric scale. A drawback associated with this approach has to do with the dynamic bias of the accelerometers which is very difficult to estimate. In [13], the information given by an altimeter is added to the monocular SLAM system to recover the metric scale.

The idea of applying cooperative approaches of SLAM to UAVs has also been explored. For example, [14,15] present a Kalman-filter-based centralized architecture. In [16–18], monocular SLAM methods for cooperative multi-UAV systems are presented to improve navigation capabilities in GPS-challenging environments. In [19], the idea of combining monocular SLAM with cooperative human–robot information to improve localization capabilities is presented. Furthermore, a single-robot SLAM approach is presented in [20], where the system state is augmented with the state of the dynamic target. In that work, robot position, map, and target are estimated using a Constrained Local Submap Filter (CLSf) based on an Extended Kalman filter (EKF) configuration. In [21], the problem of cooperative localization and target tracking with a team of moving robots is addressed. In this case, a least-squares minimization approach is followed and solved using sparse optimization. However, the main drawback of this method is related to the fact that the positions of landmarks are assumed a priori. In [22], a range-based cooperative localization method is proposed for multiple platforms with different structures. In this case, the dead reckoning system is implemented by means of an adaptive ant colony optimization particle filter algorithm. Furthermore, a method that incorporates the ultra-wideband technology into SLAM is presented in [23].

In a previous work by the authors [24], the problem of cooperative visual-SLAM based tracking of a lead agent was addressed. With big differences from the present work, where the (single robot) monocular-SLAM problem is addressed, in [24] a team of aerial robots in flight formation had to follow the dynamic lead agent. When two or more camera-robots are considered in the system, the problem of landmark initialization, as well as the problem of recovering the metric scale of the world, can be solved using a visual pseudo-stereo approach. On the other hand, the former problems can constitute a bigger challenge, if only a camera-robot is available in the system. This work deals with this last scenario.

### *1.1. Objectives and Contributions*

Recently, in [25], a visual SLAM method using an RGB-D camera was presented. In that work, the information given by the RGB-D camera is used to directly obtain depth information of its surroundings. However, the depth range of that kind of camera is quite limited. In [26], a method for the initialization

of characteristics in visual SLAM, employing the algorithm based on planar homography constraints, is presented. In that case, it is assumed that the camera only moves in a planar scene. In [27], a visual SLAM system that integrates a monocular camera and a 1D-laser range finder is presented; it seeks to provide scale recovering and drift correction. On the other hand, LiDAR-like sensors are generally expensive and can over weigh the system for certain applications presenting moving parts which can induce some errors.

Trying to present an alternative to related approaches, in this work, the use of a visual-based SLAM scheme is studied for addressing the problem of estimating the position of an aerial robot and a cooperative target in GPS-denied environments. The general idea is to use a set of a priori unknown static natural landmarks and the cooperation between a UAV and a target for locating both the aerial robot and the target moving freely in the 3D space. This objective is achieved using (i) monocular measurements of the target and the landmarks, (ii) measurements of altitude of the UAV, and (iii) range measurements between UAV and target.

The well-known EKF-SLAM methodology is used as the main estimation technique for the proposed cooperative monocular-based SLAM scheme. In this work, since the observability plays a key role in the convergence and robustness of the EKF ([28,29]), the observability properties of the system are analyzed using a nonlinear observability test. In particular, it is shown that by the sole addition of altitude measurement, the observability properties of the SLAM system are improved. In this case, the inclusion of the altimeter in monocular SLAM has been proposed previously in other works, but no such observability analyses have been done before.

In monocular-based SLAM systems, the process of initializing the new landmarks into the system state plays an important role in the performance of the system as well [30]. When only monocular measurements of landmarks are available, it is not easy to obtain 3D information from them. In this case, it becomes a difficult task to properly initialize the new map features into the system state due to the missing information. Therefore, a novel technique to estimate the approximate depth of the new visual landmarks is proposed in this work. The main idea is to take advantage of the UAV–Target cooperative scheme to infer the depth of landmarks near the target. In this case, it is shown that by the addition of altitude measurements and by the use of the proposed initialization technique, the problem of recovering the metric scale is overcome.

This work also presents a formation control scheme that allows carrying out the formation of the UAV with respect to the target. Moreover, the stability of the control system is assured utilizing the Lyapunov theory. In simulations, the state estimated by the SLAM system is used as a feedback to the proposed control scheme to test the closed-loop performance of both the estimator and the control. Finally, experiments with real data are presented to validate the applicability and performance of the proposed method.

## 1.2. Paper Outline

This work presents the following structure: mathematical models and system specifications are presented in Section 2. The nonlinear observability analysis is presented in Section 3. The proposed SLAM approach is described in Section 4. The control system is described in Section 5. Section 6 shows the results obtained from numerical simulations and with real data experiments. Finally, conclusions and final remarks of this work are given in Section 7.

## 2. System Specification

In this section, the mathematical models that will be used in this work are introduced. First, the model used for representing the dynamics of a UAV–camera system, and the model used for representing the dynamics of the target are described. Then, the model for representing the landmarks as map features is described. Furthermore, measurement models are introduced: (i) the camera projection model, (ii) the altimeter measurement model, and (iii) the range measurement model.

In applications like aerial vehicles, the attitude and heading (roll, pitch, and yaw) estimation is properly handled with available AHRS systems (e.g., [31,32]), so in this work, the estimated attitude of the vehicle is assumed to be provided by an Attitude and Heading Reference Systems (AHRS) as well as the orientation of the camera pointing always toward the ground. In practice, the foregoing assumption can be easily addressed, for instance, with the use of a servo-controlled camera gimbal or digital image stabilization (e.g., [33]). To this effect, it is important to note that the use of reliable commercial-degree AHRS and gimbal devices are assumed.

Taking into account the previous considerations, the system state can be simplified by removing the variables related to attitude and heading (which are provided by the AHRS). Therefore, the problem will be focused on the position estimation.

### 2.1. Dynamics of the System

Let consider the following continuous-time model describing the dynamics of the proposed system (see Figure 1):

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{x}}_t \\ \dot{\mathbf{v}}_t \\ \dot{\mathbf{x}}_c \\ \dot{\mathbf{v}}_c \\ \dot{\mathbf{x}}_a^i \end{bmatrix} = \begin{bmatrix} \mathbf{v}_t \\ \mathbf{0}_{3 \times 1} \\ \mathbf{v}_c \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (1)$$

where the state vector  $\mathbf{x}$  is defined as:

$$\mathbf{x} = \left[ \mathbf{x}_t \quad \mathbf{v}_t \quad \mathbf{x}_c \quad \mathbf{v}_c \quad \mathbf{x}_a^i \right]^T \quad (2)$$

with  $i = 1, \dots, n_1$ , where  $n_1$  is the number of landmarks included into the map. In this work, the term *landmarks* will be used to refer to natural features of the environment that are detected and tracked from the images acquired by a camera.

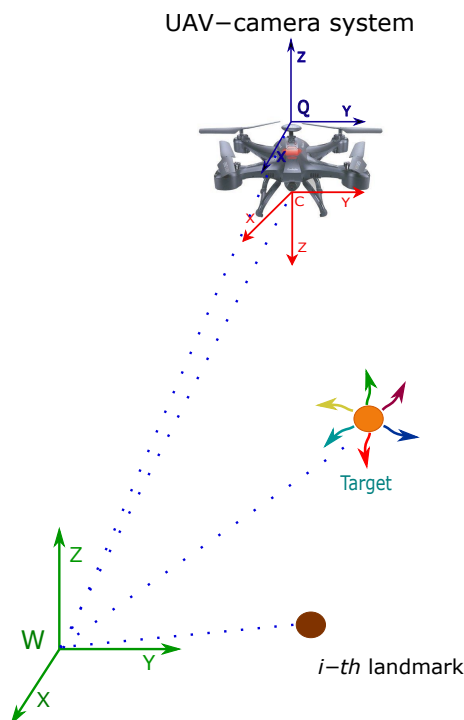


Figure 1. Coordinate reference systems.

Additionally, let  $\mathbf{x}_t = [x_t \ y_t \ z_t]^T$  represent the position (in meters) of the target, with respect to the reference system  $W$ . Let  $\mathbf{x}_c = [x_c \ y_c \ z_c]^T$  represent the position (in meters) of the reference system  $C$  of the camera, with respect to the reference system  $W$ . Let  $\mathbf{v}_t = [\dot{x}_t \ \dot{y}_t \ \dot{z}_t]^T$  represent the linear velocity (in  $\frac{m}{s}$ ) of the target. Let  $\mathbf{v}_c = [\dot{x}_c \ \dot{y}_c \ \dot{z}_c]^T$  represent the linear velocity (in  $\frac{m}{s}$ ) of the camera. Finally, let  $\mathbf{x}_a^i = [x_a^i \ y_a^i \ z_a^i]^T$  be the position of the  $i$ -th landmark (in meters) with respect to the reference system  $W$ . In Equation (1), the UAV–camera system, as well as the target, is assumed to move freely in the three-dimensional space. Let note that a non-acceleration model is assumed for the UAV–camera system and the target. Non-acceleration models are commonly used in monocular SLAM systems. In this case, it will be seen in Section 4 that unmodeled dynamics are represented by means of zero-mean Gaussian noise. In any case, augmenting the target model to consider higher-order dynamics could be straightforward. Furthermore, note that landmarks are assumed to remain static.

### 2.2. Camera Measurement Model for the Projection of Landmarks

Let consider the projection of a single landmark over the image plane of a camera. Using the pinhole model [34] the following expression can be defined:

$$\mathbf{h}_c^i = \begin{bmatrix} u_c^i \\ v_c^i \end{bmatrix} = \frac{1}{z_d^i} \begin{bmatrix} \frac{f_c}{d_u} & 0 \\ 0 & \frac{f_c}{d_v} \end{bmatrix} \begin{bmatrix} x_d^i \\ y_d^i \end{bmatrix} + \begin{bmatrix} c_u + d_{ur} + d_{ut} \\ c_v + d_{vr} + d_{vt} \end{bmatrix} \quad (3)$$

Let  $[u_c^i, v_c^i]$  define the coordinates (in pixels) of the projection of the  $i$ -th landmark over the image of the camera. Let  $f_c$  be the focal length (in meters) of the camera. Let  $[d_u, d_v]$  be the conversion parameters (in  $m/pixel$ ) for the camera. Let  $[c_u, c_v]$  be the coordinates (in pixels) of the image central point of the camera. Let  $[d_{ur}, d_{vr}]$  be components (in pixels) accounting for the radial distortion of the camera. Let  $[d_{ut}, d_{vt}]$  be components (in pixels) accounting for the tangential distortion of the camera. All the intrinsic parameters of the camera are assumed to be known using any available calibration methods. Let  $\mathbf{p}_d^i = [x_d^i \ y_d^i \ z_d^i]^T$  represent the position (in meters) of the  $i$ -th landmark with respect to the coordinate reference system  $C$  of the camera where

$$\mathbf{p}_d^i = {}^W\mathbf{R}_c(\mathbf{x}_a^i - \mathbf{x}_c) \quad (4)$$

and  ${}^W\mathbf{R}_c \in SO3$  is the rotation matrix, that transforms from the world coordinate reference system  $W$  to the coordinate reference system  $C$  of the camera. Recall that the rotation matrix  ${}^W\mathbf{R}_c$  is known and constant, by the assumption of using the servo-controlled camera gimbal.

### 2.3. Camera Measurement Model for the Projection of the Target

Let consider the projection of the target over the image plane of a camera. In this case, it is assumed that some visual feature points can be extracted from the target by means of some available computer vision algorithms like [35–38] or [39].

Using the pinhole model the following expression can be defined:

$$\mathbf{h}_c^t = \begin{bmatrix} u_c^t \\ v_c^t \end{bmatrix} = \frac{1}{z_d^t} \begin{bmatrix} \frac{f_c}{d_u} & 0 \\ 0 & \frac{f_c}{d_v} \end{bmatrix} \begin{bmatrix} x_d^t \\ y_d^t \end{bmatrix} + \begin{bmatrix} c_u + d_{ur} + d_{ut} \\ c_v + d_{vr} + d_{vt} \end{bmatrix} \quad (5)$$

Let  $\mathbf{p}_d^t = [x_d^t \ y_d^t \ z_d^t]^T$  represent the position (in meters) of the target with respect to the coordinate reference system  $C$  of the camera, and:

$$\mathbf{p}_d^t = {}^W\mathbf{R}_c(\mathbf{x}_t - \mathbf{x}_c) \quad (6)$$

#### 2.4. Altimeter Measurement Model

Let consider an altimeter carried by the UAV. Based on altimeter readings, measurements of UAV altitude are obtained, therefore this model is simply defined by:

$$h_a = z_c \quad (7)$$

It is important to note that the only strict requirement for the proposed method is the availability of altitude measurements respect to the reference system  $W$ . In this case, the typical barometer-based altimeters which are equipped in most UAVs can be configured to provide such kind of measurement [40].

#### 2.5. Range Measurement Model

Let consider the availability of a range sensor. Its measurements of the relative distance of the UAV with respect to the target are obtained. In this case, the measurement model is defined by:

$$h_r = \sqrt{(x_t - x_c)^2 + (y_t - y_c)^2 + (z_t - z_c)^2} \quad (8)$$

For practical implementation, several techniques like [41] or [42] can be used to obtain these kinds of measurements. On the other hand, a practical limitation for using these techniques is the requirement of a target equipped with such a device. Thus, the application of the proposed method with non-cooperative targets becomes more challenging.

### 3. Observability Analysis

In this section, the nonlinear observability properties of the proposed system are studied. Observability is an inherent property of a dynamic system and has an important role in the accuracy and stability of its estimation process. Moreover, this fact has important consequences in the convergence of the EKF-based SLAM.

In particular, it will be shown that the inclusion of the altimeter measurements improves the observability properties of the SLAM system.

A system is defined as observable if the initial state  $\mathbf{x}_0$ , at any initial time  $t_0$ , can be determined given the state transition model  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ , the observation model  $\mathbf{y} = \mathbf{h}(\mathbf{x})$ , and observations  $\mathbf{z}[t_0, t]$  from time  $t_0$  to a finite time  $t$ . Given the observability matrix  $\mathcal{O}$ , a non-linear system is *locally weakly observable* if the condition  $\text{rank}(\mathcal{O}) = \text{dim}(\mathbf{x})$  is verified [43].

#### 3.1. Observability Matrix

An observability matrix  $\mathcal{O}$  can be constructed in the following manner:

$$\mathcal{O} = \left[ \begin{array}{cccccccc} \frac{\partial(\mathcal{L}_f^0(\mathbf{h}_c^i))}{\partial \mathbf{x}} & \frac{\partial(\mathcal{L}_f^1(\mathbf{h}_c^i))}{\partial \mathbf{x}} & \dots & \frac{\partial(\mathcal{L}_f^0(\mathbf{h}_c^t))}{\partial \mathbf{x}} & \frac{\partial(\mathcal{L}_f^1(\mathbf{h}_c^t))}{\partial \mathbf{x}} & \frac{\partial(\mathcal{L}_f^0(h_a))}{\partial \mathbf{x}} & \frac{\partial(\mathcal{L}_f^1(h_a))}{\partial \mathbf{x}} & \frac{\partial(\mathcal{L}_f^0(h_r))}{\partial \mathbf{x}} & \frac{\partial(\mathcal{L}_f^1(h_r))}{\partial \mathbf{x}} \end{array} \right]^T \quad (9)$$

where  $\mathcal{L}_f^s \mathbf{h}$  represent the  $s$ -th-order Lie derivative [44], of the scalar field  $\mathbf{h}$  respect to the vector field  $\mathbf{f}$ . In this work, the rank calculation of Equation (9) was carried out numerically using MATLAB. The degree of Lie derivatives, used for computing  $\mathcal{O}$ , was determined by gradually augmenting the matrix  $\mathcal{O}$  with higher-order derivatives until its rank remained constant. Based on this approach, only Lie derivatives of zero and first order were needed to construct the observability matrix for all the cases.

The description of the zero and first order Lie derivatives used for constructing Equation (9) are presented in Appendix A. Using these derivatives the observability matrix in Equation (9) can be expanded as follows:

$$\mathcal{O} = \left[ \begin{array}{cc|cc|cc} \mathbf{0}_{2 \times 6} & & -\mathbf{H}_c^i \mathbf{W} \mathbf{R}_c & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3(i-1)} & \mathbf{H}_c^i \mathbf{W} \mathbf{R}_c & \mathbf{0}_{2 \times 3(n_1-i)} \\ \mathbf{0}_{2 \times 6} & & \mathbf{H}_{dc}^i & -\mathbf{H}_c^i \mathbf{W} \mathbf{R}_c & \mathbf{0}_{2 \times 3(i-1)} & -\mathbf{H}_{dc}^i & \mathbf{0}_{2 \times 3(n_1-i)} \\ \vdots & & \vdots & & & \vdots & \\ \mathbf{H}_c^t \mathbf{W} \mathbf{R}_c & \mathbf{0}_{2 \times 3} & -\mathbf{H}_c^t \mathbf{W} \mathbf{R}_c & \mathbf{0}_{2 \times 3} & & \mathbf{0}_{2 \times 3n_1} & \\ -\mathbf{H}_{dc}^t & \mathbf{H}_c^t \mathbf{W} \mathbf{R}_c & \mathbf{H}_{dc}^t & -\mathbf{H}_c^t \mathbf{W} \mathbf{R}_c & & \mathbf{0}_{2 \times 3n_1} & \\ \mathbf{0}_{1 \times 6} & & \left[ \mathbf{0}_{1 \times 2} \quad 1 \right] & \mathbf{0}_{1 \times 3} & & \mathbf{0}_{1 \times 3n_1} & \\ \mathbf{0}_{1 \times 6} & & \mathbf{0}_{1 \times 3} & \left[ \mathbf{0}_{1 \times 2} \quad 1 \right] & & \mathbf{0}_{1 \times 3n_1} & \\ \mathbf{H}_r & \mathbf{0}_{1 \times 3} & -\mathbf{H}_r & \mathbf{0}_{1 \times 3} & & \mathbf{0}_{1 \times 3n_1} & \\ \mathbf{H}_{dr} & \mathbf{H}_r & -\mathbf{H}_{dr} & -\mathbf{H}_r & & \mathbf{0}_{1 \times 3n_1} & \end{array} \right] \quad (10)$$

In Equations (9) and (10), Lie derivatives that belong to each kind of measurement are distributed as: first two rows (or first two elements in Equation (9)) are for monocular measurements of the landmarks; second two rows (or second two elements) are for monocular measurements of the target; third two rows (or third two elements) are for altitude measurements; and last two rows (or last two elements) are for range (UAV–target) measurements.

### 3.2. Theoretical Results

Two different cases of system configurations were analyzed. The idea is to study how the observability of the system is affected due to the availability (or unavailability) of the altimeter measurements.

#### 3.2.1. without Altimeter Measurements

In this case, considering only the respective derivatives on the observability matrix in Equation (10), the maximum rank of the observability matrix  $\mathcal{O}$  is  $\text{rank}(\mathcal{O}) = (3n_1 + 12) - 4$ . In this case,  $n_1$  is the number of measured landmarks, 12 is the number of states of the UAV–camera system and the target, and 3 is the number of states per landmark. Therefore,  $\mathcal{O}$  will be rank deficient ( $\text{rank}(\mathcal{O}) < \text{dim}(\mathbf{x})$ ). The unobservable modes are spanned by the right nullspace basis  $\mathbf{N}_1$  of the observability matrix  $\mathcal{O}$ .

It is straightforward to verify that the right nullspace basis of  $\mathcal{O}$  spans for  $\mathbf{N}_1$ , (i.e.,  $\mathcal{O}\mathbf{N}_1 = \mathbf{0}$ ). From Equation (11) it can be seen that the unobservable modes cross through all states, and thus all states are unobservable. It should be noted that adding Lie derivatives of higher-order to the observability matrix the previous result does not improve.



$$\mathbf{N}_1 = null(\mathcal{O}) = \left( \frac{1}{z_a^{(i-1)} - z_a^i} \right) \left[ \begin{array}{ccc|ccc}
 \mathbf{x}_c - \mathbf{x}_a^i & \begin{bmatrix} (z_a^{(i-1)} - z_a^i) \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ (z_a^{(i-1)} - z_a^i) \\ 0 \end{bmatrix} & - & \begin{bmatrix} x_c - x_a^i \\ y_c - y_a^i \\ z_c - z_a^{(i-1)} \end{bmatrix} \\
 \mathbf{v}_c & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & & -\mathbf{v}_c \\
 \hline
 \mathbf{x}_c - \mathbf{x}_a^i & \begin{bmatrix} (z_a^{(i-1)} - z_a^i) \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ (z_a^{(i-1)} - z_a^i) \\ 0 \end{bmatrix} & - & \begin{bmatrix} x_c - x_a^i \\ y_c - y_a^i \\ z_c - z_a^{(i-1)} \end{bmatrix} \\
 \mathbf{v}_c & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & & -\mathbf{v}_c \\
 \hline
 \mathbf{x}_a^1 - \mathbf{x}_a^i & \vdots & \vdots & - & \begin{bmatrix} x_a^1 - x_a^i \\ y_a^1 - y_a^i \\ z_a^1 - z_a^{(i-1)} \end{bmatrix} \\
 \vdots & \begin{bmatrix} (z_a^{(i-1)} - z_a^i) \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ (z_a^{(i-1)} - z_a^i) \\ 0 \end{bmatrix} & & \vdots \\
 \mathbf{x}_a^{(i-2)} - \mathbf{x}_a^i & \vdots & \vdots & - & \begin{bmatrix} x_a^{(i-2)} - x_a^i \\ y_a^{(i-2)} - y_a^i \\ z_a^{(i-2)} - z_a^{(i-1)} \end{bmatrix} \\
 \mathbf{x}_a^{(i-1)} - \mathbf{x}_a^i & \begin{bmatrix} (z_a^{(i-1)} - z_a^i) \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ (z_a^{(i-1)} - z_a^i) \\ 0 \end{bmatrix} & - & \begin{bmatrix} x_a^{(i-1)} - x_a^i \\ y_a^{(i-1)} - y_a^i \\ 0 \end{bmatrix} \\
 \mathbf{0}_{3 \times 1} & \vdots & \vdots & & \begin{bmatrix} 0 \\ 0 \\ (z_a^{(i-1)} - z_a^i) \end{bmatrix}
 \end{array} \right] \quad (11)$$

### 3.2.2. with Altimeter Measurements

When altimeter measurements are taking into account, the observability matrix in Equation (10) is rank deficient ( $rank(\mathcal{O}) < dim(\mathbf{x})$ ), with  $rank(\mathcal{O}) = (3n_1 + 12) - 2$ . In such a case, the following right nullspace basis  $\mathbf{N}_2$  spans the unobservable modes of  $\mathcal{O}$ :

$$\mathbf{N}_2 = null(\mathcal{O}) = \left[ \begin{array}{cccc|cccc|cccc|cccc}
 [1 \ 0 \ 0]^T & \mathbf{0}_{3 \times 1} & [1 \ 0 \ 0]^T & \mathbf{0}_{3 \times 1} & [1 \ 0 \ 0]^T & \dots & [1 \ 0 \ 0]^T & \\
 [0 \ 1 \ 0]^T & \mathbf{0}_{3 \times 1} & [0 \ 1 \ 0]^T & \mathbf{0}_{3 \times 1} & [0 \ 1 \ 0]^T & \dots & [0 \ 1 \ 0]^T & 
 \end{array} \right]^T \quad (12)$$

It can be verified that the right nullspace basis of  $\mathcal{O}$  spans for  $\mathbf{N}_2$ , (i.e.,  $\mathcal{O}\mathbf{N}_2 = \mathbf{0}$ ). From Equation (12) it can be observed that the unobservable modes are related to the global position in  $x$  and  $y$  of the UAV-camera system, the landmarks, and the target. In this case, the rest of the states are observable. It should be noted that adding Lie derivatives of higher-order to the observability matrix the previous result does not improve.



Table 1. Results of observability test.

	Unobservable Modes	Unobservable States	Observable States
Without altimeter measurements	4	$\mathbf{x}$	-
With altimeter measurements	2	$x_t, y_t, x_c, y_c, x_a^i, y_a^i$	$z_t, z_c, z_a^i, \mathbf{v}_t, \mathbf{v}_c$

### 3.2.3. Remarks on the Theoretical Results

To interpret the former theoretical results, it is important to recall that any world-centric SLAM system is partially observable in the absence of global measurements (e.g., GPS measurements).

In this case, the SLAM system computes the position and velocity within its map, and not respect to the global reference system. Fortunately, this is not a problem for some applications that require local or relative position estimates, for instance the problem addressed in this work that implies to following a moving target.

On the other hand, it is worth noting how the simple inclusion of an altimeter improves the observability properties of the system when GPS measurements are not considered (see Table 1). It is very interesting to observe that, besides the states along the z-axis [ $z_t, z_c, z_a^i$ ] (as one could expect), the velocity of the camera-robot (which is global-referenced) becomes observable when altitude measurements are included. In this case, since the target is estimated respect to the camera, the global velocity of the target becomes observable.

Accordingly, it is also important to note that, since the range and monocular measurements to the target are used only for estimating the position of the target with respect to the camera-robot, these measurements affect neither the observability of the camera-robot state nor the observability of the landmarks states.

In other words, the target measurements create only a “link” to the camera-robot state that allows estimating the relative position of the target but does not provide any information about the state of the camera-robot, and for this reason, they are not included in the observability analysis.

Later, it will be seen how the target position is used for improving the initialization of nearby landmarks, which in turn improves the robustness and accuracy of the system.

A final but very important remark is to consider that the order of Lie derivatives and the rank calculation of Equation (9) were determined numerically, but not analytically. Therefore, there is still a chance, in rigorous terms, that a subset of the unobservable states determined by the analysis is in reality observable (see [43]).

## 4. EKF-Based Slam

In this work, the standard EKF-based SLAM scheme [45,46] is used to estimate the system state in Equation (2). The architecture of the proposed system is shown in Figure 2.

From Equation (1), the following discrete system can be defined:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{n}_{k-1}) = \begin{bmatrix} \mathbf{x}_{tk} \\ \mathbf{v}_{tk} \\ \mathbf{x}_{ck} \\ \mathbf{v}_{ck} \\ \mathbf{e}\mathbf{x}_{ak}^j \\ \mathbf{p}\mathbf{x}_{ak}^n \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{tk-1} + (\mathbf{v}_{tk-1})\Delta t \\ \mathbf{v}_{tk-1} + \boldsymbol{\eta}_{tk-1} \\ \mathbf{x}_{ck-1} + (\mathbf{v}_{ck-1})\Delta t \\ \mathbf{v}_{ck-1} + \boldsymbol{\eta}_{ck-1} \\ \mathbf{e}\mathbf{x}_{ak-1}^j \\ \mathbf{p}\mathbf{x}_{ak-1}^n \end{bmatrix} \quad (13)$$

$$\mathbf{n}_k = \begin{bmatrix} \boldsymbol{\eta}_{tk} \\ \boldsymbol{\eta}_{ck} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_t \Delta t \\ \mathbf{a}_c \Delta t \end{bmatrix} \quad (14)$$

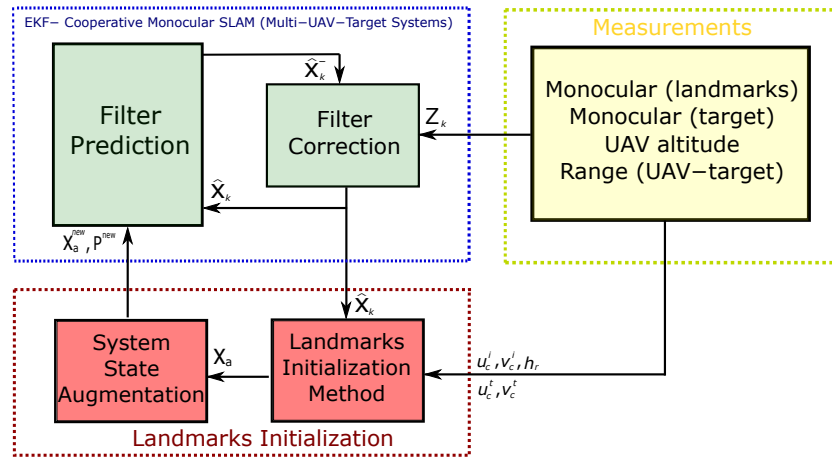


Figure 2. Block diagram showing the EKF-SLAM architecture of the proposed system.

From Equations (3), (5), (7) and (8), the system measurements are defined as follows:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{r}_k) = \begin{bmatrix} \mathbf{e}\mathbf{h}_{c_k}^j + \mathbf{e}\mathbf{r}_{c_k}^j \\ \mathbf{p}\mathbf{h}_{c_k}^n + \mathbf{p}\mathbf{r}_{c_k}^n \\ \mathbf{h}_{t_k} + \mathbf{r}_{t_k} \\ h_{a_k} + r_{a_k} \\ h_{r_k} + r_{r_k} \end{bmatrix} \quad (15)$$

$$\mathbf{r}_k = \begin{bmatrix} \mathbf{e}\mathbf{r}_{c_k}^j \\ \mathbf{p}\mathbf{r}_{c_k}^n \\ \mathbf{r}_{t_k} \\ r_{a_k} \\ r_{r_k} \end{bmatrix} \quad (16)$$

Let  $\mathbf{e}\mathbf{x}_a^j = [e x_a^j \ e y_a^j \ e z_a^j]^\top$  be the  $j$ -th landmark defined by its Euclidean parametrization. Let  $\mathbf{p}\mathbf{x}_a^n = [p x_{c_0}^n \ p y_{c_0}^n \ p z_{c_0}^n \ p\theta_a^n \ p\phi_a^n \ p\rho_a^n]^\top$  be the  $n$ -th landmark defined by its inverse of the depth parametrization,  $j = 1, \dots, n_2$ , where  $n_2$  is the number of landmarks with Euclidean parametrization,  $n = 1, \dots, n_3$ , where  $n_3$  is the number of landmarks with inverse of the depth parametrization, and  $n_1 = n_2 + n_3$ .

Let  ${}^p\mathbf{x}_{c_0}^n = [{}^p x_{c_0}^n \ {}^p y_{c_0}^n \ {}^p z_{c_0}^n]^\top$  represent the position (in meters) of the camera when the feature  $\mathbf{p}\mathbf{x}_a^n$  was observed for the first time. Let  ${}^p\theta_a^n$  and  ${}^p\phi_a^n$  be azimuth and elevation respectively (respect to the global reference frame  $W$ ). Let  ${}^p\rho_a^n = \frac{1}{{}^p d^n}$  be the inverse of the depth  ${}^p d^n$ . Let  $\mathbf{e}\mathbf{h}_c^j$  be the projection over the image plane of a camera of the  $j$ -th landmark. Let  $\mathbf{p}\mathbf{h}_c^n$  be the projection over the image plane of a camera of the  $n$ -th landmark.

In Equation (14),  $\mathbf{a}_t$  and  $\mathbf{a}_c$  are zero-mean Gaussian noise representing unknown linear accelerations dynamics. Moreover,  $\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ ,  $\mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$  are uncorrelated noise vectors affecting respectively the system dynamics and the system measurements. Let  $k$  be the sample step, and  $\Delta t$  is the time differential. It is important to note that the proposed scheme does not depend on a specific aircraft dynamical model.

The EKF prediction equations are:

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}, \mathbf{0}) \quad (17)$$

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^\top + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^\top \quad (18)$$

The EKF update equations are:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, \mathbf{0})) \quad (19)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_k^- \quad (20)$$

with

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T)^{-1} \quad (21)$$

and

$$\begin{aligned} \mathbf{A}_k &= \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{k-1}, \mathbf{0}) & \mathbf{C}_k &= \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_k^-, \mathbf{0}) \\ \mathbf{W}_k &= \frac{\partial \mathbf{f}}{\partial \mathbf{n}}(\hat{\mathbf{x}}_{k-1}, \mathbf{0}) & \mathbf{V}_k &= \frac{\partial \mathbf{h}}{\partial \mathbf{r}}(\hat{\mathbf{x}}_k^-, \mathbf{0}) \end{aligned} \quad (22)$$

Let  $\mathbf{K}$  be the Kalman gain, and let  $\mathbf{P}$  be the system covariance matrix.

#### 4.1. Map Features Initialization

The system state  $\mathbf{x}$  is augmented with new map features when a landmark is observed for the first time. The landmark can be initialized in one of two different parametrizations: (i) Euclidean parametrization and (ii) Inverse depth parametrization, depending on how close this landmark is from the target. Since the target is assumed to move over the ground, the general idea is to use the range information provided by the target to infer the initial depth of the landmarks near to the target. In this case, it will be assumed that the landmarks near the target lie at a similar altitude, situation encountered in most of the applications. It is important to recall that the initialization of landmarks plays a fundamental role in the robustness and convergence of the EKF-based SLAM.

##### 4.1.1. Initialization of Landmarks near to the Target

A landmark is initialized with a Euclidean parameterization if it is supposed arbitrarily near the target. This assumption is made if the landmark is within a selected area of the image (see Section 4.1.4). In this case, the landmark can be initialized with the information given by the range measurement between the UAV and the target, which is assumed to be approximately equal to the depth that the landmark has respect to the camera.

Therefore, the following equation is defined:

$${}^e \mathbf{x}_a^j = \hat{\mathbf{x}}_{c_k} + h_r \begin{bmatrix} \cos({}^e \theta_a^j) \cos({}^e \phi_a^j) \\ \sin({}^e \theta_a^j) \cos({}^e \phi_a^j) \\ \sin({}^e \phi_a^j) \end{bmatrix} \quad (23)$$

where  $\hat{\mathbf{x}}_{c_k}$  is the estimated position of the camera when the feature  ${}^e \mathbf{x}_a^j$  was observed for first time, and

$$\begin{bmatrix} {}^e \theta_a^j \\ {}^e \phi_a^j \end{bmatrix} = \begin{bmatrix} \arctan 2 \left( {}^e g_{a_y}^j, {}^e g_{a_x}^j \right) \\ \arctan 2 \left( {}^e g_{a_z}^j, \sqrt{\left( {}^e g_{a_x}^j \right)^2 + \left( {}^e g_{a_y}^j \right)^2} \right) \end{bmatrix} \quad (24)$$

with  ${}^e\mathbf{g}_a^j = [{}^e g_{ax}^j \quad {}^e g_{ay}^j \quad {}^e g_{az}^j]^T = \mathbf{W}\mathbf{R}_c^T [{}^e u_c^j \quad {}^e v_c^j \quad f_c]^T$ . Where,  $[{}^e u_c^j, {}^e v_c^j]$  define the coordinates (in pixels) of the projection of the  $j$ -th landmark over the image of the camera. In case of a landmark with Euclidean parameterization, the projection over the image plane of a camera is defined:

$${}^e\mathbf{h}_c^j = \begin{bmatrix} {}^e u_c^j \\ {}^e v_c^j \end{bmatrix} = \frac{1}{{}^e z_d^j} \begin{bmatrix} \frac{f_c}{d_u} & 0 \\ 0 & \frac{f_c}{d_v} \end{bmatrix} \begin{bmatrix} {}^e x_d^j \\ {}^e y_d^j \end{bmatrix} + \begin{bmatrix} c_u + d_{ur} + d_{ut} \\ c_v + d_{vr} + d_{vt} \end{bmatrix} \quad (25)$$

with

$${}^e\mathbf{p}_d^j = \begin{bmatrix} {}^e x_d^j \\ {}^e y_d^j \\ {}^e z_d^j \end{bmatrix} = \mathbf{W}\mathbf{R}_c ({}^e\mathbf{x}_a^j - \mathbf{x}_c) \quad (26)$$

#### 4.1.2. Initialization of Landmarks Far from the Target

A landmark is initialized with an inverse depth parametrization if it is supposed arbitrarily far from the target. This assumption is made if the landmark is outside a selected area of the image (see Section 4.1.4). In this case,  ${}^p\mathbf{x}_{c_0}^n$  is given for the estimated position of the camera  $\hat{\mathbf{x}}_{c_k}$  when the feature  ${}^p\mathbf{x}_a^n$  was observed for the first time. Furthermore, the following equation is defined:

$$\begin{bmatrix} {}^p\theta_a^n \\ {}^p\phi_a^n \end{bmatrix} = \begin{bmatrix} \arctan 2 \left( \frac{{}^p g_{ay}^n, {}^p g_{ax}^n}{{}^p g_{az}^n, \sqrt{({}^p g_{ax}^n)^2 + ({}^p g_{ay}^n)^2}} \right) \\ \arctan 2 \left( \frac{{}^p g_{ay}^n, {}^p g_{ax}^n}{{}^p g_{az}^n, \sqrt{({}^p g_{ax}^n)^2 + ({}^p g_{ay}^n)^2}} \right) \end{bmatrix} \quad (27)$$

with  ${}^p\mathbf{g}_a^n = [{}^p g_{ax}^n \quad {}^p g_{ay}^n \quad {}^p g_{az}^n]^T = \mathbf{W}\mathbf{R}_c^T [{}^p u_c^n \quad {}^p v_c^n \quad f_c]^T$ . Where,  $[{}^p u_c^n, {}^p v_c^n]$  define the coordinates (in pixels) of the projection of the  $n$ -th landmark over the image of the camera.  ${}^p\rho_a^n$  is initialized as it is shown in [47]. In case of a landmark with inverse depth parametrization, the projection over the image plane of a camera is defined by:

$${}^p\mathbf{h}_c^n = \begin{bmatrix} {}^p u_c^n \\ {}^p v_c^n \end{bmatrix} = \frac{1}{{}^p z_d^n} \begin{bmatrix} \frac{f_c}{d_u} & 0 \\ 0 & \frac{f_c}{d_v} \end{bmatrix} \begin{bmatrix} {}^p x_d^n \\ {}^p y_d^n \end{bmatrix} + \begin{bmatrix} c_u + d_{ur} + d_{ut} \\ c_v + d_{vr} + d_{vt} \end{bmatrix} \quad (28)$$

with

$${}^p\mathbf{p}_d^n = \begin{bmatrix} {}^p x_d^n \\ {}^p y_d^n \\ {}^p z_d^n \end{bmatrix} = \mathbf{W}\mathbf{R}_c \left( {}^p\mathbf{x}_{c_0}^n + \frac{1}{{}^p\rho_a^n} \begin{bmatrix} \cos({}^p\theta_a^n) \cos({}^p\phi_a^n) \\ \sin({}^p\theta_a^n) \cos({}^p\phi_a^n) \\ \sin({}^p\phi_a^n) \end{bmatrix} - \mathbf{x}_c \right) \quad (29)$$

#### 4.1.3. State Augmentation

To initialize a new landmark, the system state  $\mathbf{x}$  must be augmented by  $\mathbf{x} = [\mathbf{x}_t \quad \mathbf{v}_t \quad \mathbf{x}_c \quad \mathbf{v}_c \quad {}^e\mathbf{x}_a^j \quad {}^p\mathbf{x}_a^n \quad \mathbf{x}_a^{new}]^T$ , being  $\mathbf{x}_a^{new}$  the new landmark which is initialized by either the Euclidean or the inverse depth parametrization. Thus, a new covariance matrix  $\mathbf{P}_{new}$  is computed by:

$$\mathbf{P}_{new} = \Delta\mathbf{J} \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^i \end{bmatrix} \Delta\mathbf{J}^T \quad (30)$$

where  $\mathbf{R}^i$  is the measurement noise covariance matrix,  $\Delta\mathbf{J}$  is the Jacobian  $\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}$ , and  $\mathbf{h}(\mathbf{x})$  is the landmark initialization function.

#### 4.1.4. Landmarks Selection Method

To determine whether a landmark is initialized with Euclidean or inverse depth parametrization, it should be determined arbitrarily if the landmark is considered near or far from the target. To achieve

this objective the following heuristic is used (see Figure 3): (1) firstly, a spherical area centered on the target of radius  $r_w$  is defined; (2) then, the radius  $r_c$  of the projected spherical area in the camera is estimated; and (3) the landmarks whose projection in the camera are within the projected spherical area ( $^i d_t \leq r_c$ ) are considered near to the target and thus, they are initialized with Euclidean parameterization (see Section 4.1.1). Otherwise ( $^i d_t > r_c$ ), landmarks are considered far from the target, and they are initialized with inverse depth parametrization (see Section 4.1.2) where  $^i d_t = \sqrt{(u_c^t - u_c^i)^2 + (v_c^t - v_c^i)^2}$ .

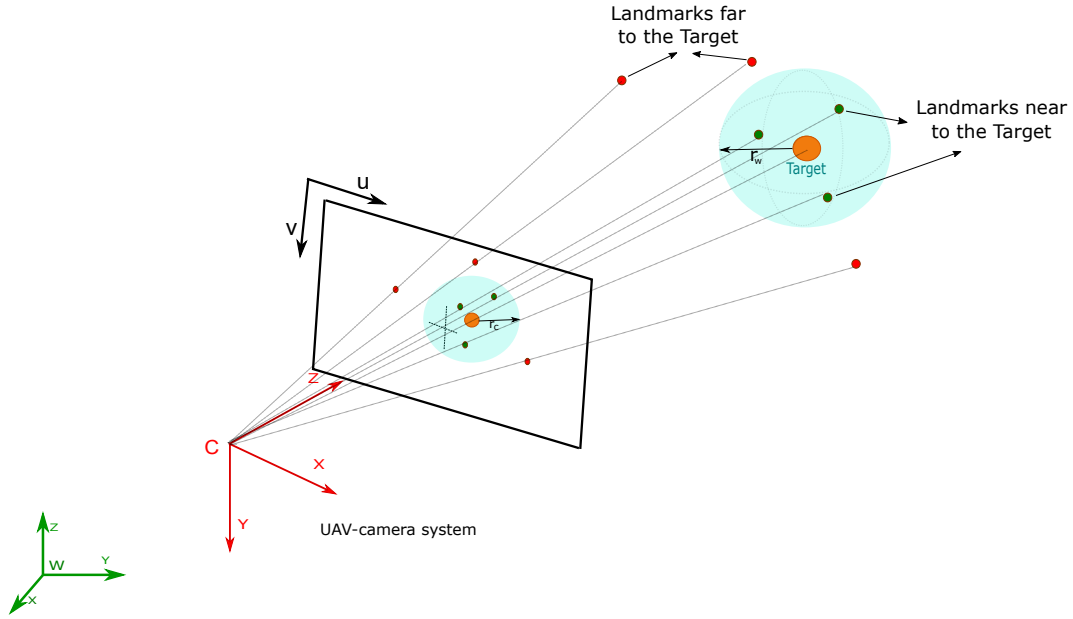


Figure 3. Landmarks selection method.

Here,  $r_c$  is estimated as follows:

$$r_c = \sqrt{(u_c^t - u_c^i)^2 + (v_c^t - v_c^i)^2} \quad (31)$$

where

$$\begin{bmatrix} u_c^r \\ v_c^r \end{bmatrix} = \frac{1}{z_d^r} \begin{bmatrix} \frac{f_c}{d_u} & 0 \\ 0 & \frac{f_c}{d_v} \end{bmatrix} \begin{bmatrix} x_d^r \\ y_d^r \end{bmatrix} + \begin{bmatrix} c_u + d_{ur} + d_{ut} \\ c_v + d_{vr} + d_{vt} \end{bmatrix} \quad (32)$$

with

$$\mathbf{p}_d^r = \begin{bmatrix} x_d^r \\ y_d^r \\ z_d^r \end{bmatrix} = {}^w \mathbf{R}_c (\hat{\mathbf{x}}_t + \boldsymbol{\eta} - \hat{\mathbf{x}}_c) \quad (33)$$

and  $\boldsymbol{\eta} = [r_w \ 0 \ 0]^T$ .

## 5. Target Tracking Control

To allow a UAV to follow a target a high-level control scheme is presented. Firstly, the kinematic model of the UAV is defined:

$$\begin{aligned}\dot{x}_q &= v_x \cos(\psi_q) - v_y \sin(\psi_q) \\ \dot{y}_q &= v_x \sin(\psi_q) + v_y \cos(\psi_q) \\ \dot{z}_q &= v_z \\ \dot{\psi}_q &= \omega\end{aligned}\quad (34)$$

Let  $\mathbf{x}_q = [x_q \ y_q \ z_q]^T$  represent the UAV position respect to the reference system  $W$  (in m). Let  $(v_x, v_y)$  represent the UAV linear velocity along the  $x$  and  $y$  axis (in m/s) respect to the reference system  $Q$ . Let  $v_z$  represent the UAV linear velocity along the  $z$  (vertical) axis (in m/s) respect to the reference system  $W$ . Let  $\psi$  represent the UAV yaw angle respect to  $W$  (in radians); and let  $\omega$  (in radians/s) is the first derivative of  $\psi$  (angular velocity).

The proposed high-level control scheme is intended to maintain a stable flight formation of the UAV with respect to the target, by generating velocity commands to the UAV. In this case, it is assumed that a low-level (i.e., actuator-level) velocity control scheme exists, like [48] or [49], that drives the velocities  $[v_x, v_y, v_z, \omega]$  commanded by a high-level control.

### 5.1. Visual Servoing and Altitude Control

By deriving Equation (5) the following expression can be obtained, neglecting the dynamics of the tangential and radial distortion components, taking into account that  $\mathbf{x}_q = \mathbf{x}_c - {}^q\mathbf{d}_c$ , where  ${}^q\mathbf{d}_c$  is the translation vector (in meters) from the coordinate reference system  $Q$  to the coordinate reference system  $C$ , and assuming  ${}^q\mathbf{d}_c$  to be known and constant:

$$\begin{bmatrix} \dot{u}_c^t \\ \dot{v}_c^t \end{bmatrix} = \mathbf{J}_c^t \mathbf{W} \mathbf{R}_c (\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_q) \quad (35)$$

with

$$\mathbf{J}_c^t = \begin{bmatrix} \frac{f_c}{d_u z_d^t} & 0 & -\frac{(u_c^t - c_u - d_{ur} - d_{ut})}{z_d^t} \\ 0 & \frac{f_c}{d_v z_d^t} & -\frac{(v_c^t - c_v - d_{vr} - d_{vt})}{z_d^t} \end{bmatrix} \quad (36)$$

Furthermore, an altitude differential  $\lambda_z$  to be maintained from the UAV to the target is defined:

$$\lambda_z = z_q - z_t \quad (37)$$

Now, differentiating Equation (37):

$$\dot{\lambda}_z = \dot{z}_q - \dot{z}_t \quad (38)$$

Taking Equations (34), (35) and (38), the following dynamics is defined:

$$\dot{\boldsymbol{\lambda}} = \mathbf{g} + \mathbf{B}\mathbf{u} \quad (39)$$

where

$$\boldsymbol{\lambda} = \begin{bmatrix} u_c^t \\ v_c^t \\ \lambda_z \\ \psi_q \end{bmatrix} \quad \mathbf{g} = \begin{bmatrix} \mathbf{J}_c^t \mathbf{W} \mathbf{R}_c & \mathbf{0}_{2 \times 1} \\ \mathbf{c}_1 & \mathbf{0}_{2 \times 1} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_t \\ 0 \end{bmatrix} \quad \mathbf{B} = - \begin{bmatrix} \mathbf{J}_c^t \mathbf{W} \mathbf{R}_c & \mathbf{0}_{2 \times 1} \\ \mathbf{c}_1 & \mathbf{c}_2 \end{bmatrix} \boldsymbol{\Omega} \quad \mathbf{u} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega \end{bmatrix} \quad (40)$$

with

$$\mathbf{c}_1 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{c}_2 = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \mathbf{\Omega} = \begin{bmatrix} \cos(\psi_q) & -\sin(\psi_q) & 0 & 0 \\ \sin(\psi_q) & \cos(\psi_q) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (41)$$

It will be assumed that disturbances, as well as unmodeled uncertainty, enters the system through the input. In this case, Equation (39) is redefined as:

$$\dot{\lambda} = \mathbf{g} + \Delta_{\mathbf{g}} + \mathbf{B}\mathbf{u} \quad (42)$$

where the term  $\Delta_{\mathbf{g}}$  (representing the unknown disturbances and uncertainties) satisfies  $\|\Delta_{\mathbf{g}}\| \leq \epsilon$ , where  $\epsilon$  is a positive constant, so it is assumed to be bounded. Based on the dynamics in Equation (39), a robust controller is designed using the sliding mode control technique [50]. For the controller, the state-feedback is obtained from the SLAM estimator presented in Section 4. In this case, it is assumed that the UAV yaw angle is obtained directly from the AHRS device. The architecture of closed-loop system is show in Figure 4.

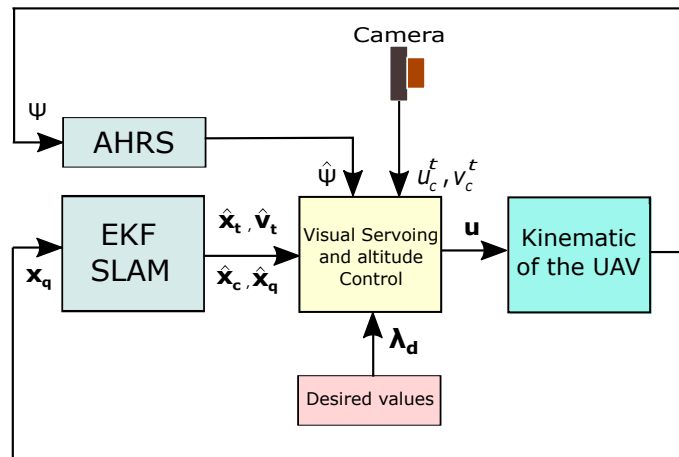


Figure 4. Control scheme.

First, the transformation  $\hat{\mathbf{x}}_q = \hat{\mathbf{x}}_c - \mathbf{q}\mathbf{d}_c^j$  is defined, to obtain the UAV estimated position in terms of the reference system  $Q$ .

To design the controller, the following expression is defined:

$$s_\lambda = e_\lambda + \mathbf{K}_1 \int_0^t e_\lambda dt \quad (43)$$

where  $\mathbf{K}_1$  is a positive definite diagonal matrix, and  $e_\lambda = \hat{\lambda} - \lambda_d$ , and  $\lambda_d$  is the reference signal vector. By deriving Equation (43) and substituting in Equations (39), the following expression is obtained:

$$\dot{s}_\lambda = -\dot{\lambda}_d + \mathbf{K}_1 e_\lambda + \hat{\mathbf{g}} + \Delta_{\mathbf{g}} + \hat{\mathbf{B}}\mathbf{u} \quad (44)$$

For the former dynamics, the following control law is proposed:

$$\mathbf{u} = \hat{\mathbf{B}}^{-1} \left( \dot{\lambda}_d - \mathbf{K}_1 e_\lambda - \hat{\mathbf{g}} - \mathbf{K}_2 \text{sign}(s_\lambda) \right) \quad (45)$$

where  $\mathbf{K}_2$  is a positive definite diagonal matrix. Appendix B shows the proof of the existence of  $\hat{\mathbf{B}}^{-1}$ .

A Lyapunov candidate function is defined to prove the stability of the closed-loop dynamics:



$$V_\lambda = \frac{1}{2} s_\lambda^T s_\lambda \quad (46)$$

with its corresponding derivative:

$$\dot{V}_\lambda = s_\lambda^T \dot{s}_\lambda = s_\lambda^T \left( -\dot{\lambda}_d + \mathbf{K}_1 e_\lambda + \hat{\mathbf{g}} + \Delta_g + \hat{\mathbf{B}} \mathbf{u} \right) \quad (47)$$

So, by substituting Equation (45) in Equation (47), the following expression can be obtained:

$$\begin{aligned} \dot{V}_\lambda &= s_\lambda^T (\Delta_g - \mathbf{K}_2 \text{sign}(s_\lambda)) \\ &\leq \| s_\lambda \| \| \Delta_g \| - s_\lambda^T \| \mathbf{K}_2 \| \text{sign}(s_\lambda) \\ &\leq \| s_\lambda \| \epsilon - \alpha s_\lambda^T \text{sign}(s_\lambda) \\ &\leq \| s_\lambda \| \epsilon - \| s_\lambda \| \alpha \\ &\leq \| s_\lambda \| (\epsilon - \alpha) \end{aligned} \quad (48)$$

where  $\alpha = \lambda_{\min}(\mathbf{K}_2)$ . Therefore, if  $\alpha$  is chosen such that  $\alpha > \epsilon$ , then  $\dot{V}_\lambda$  will be negative definite. In this case, the dynamics defining the flight formation will reach the surface  $s_\lambda = 0$  and will remain there in a finite time.

## 6. Experimental Results

To validate the performance of the proposed method, simulations and experiments with real data have been carried out.

### 6.1. Simulations

#### 6.1.1. Simulation Setup

The proposed cooperative UAV–Target visual-SLAM method is validated through computer simulations. For this purpose, a Matlab<sup>®</sup> implementation of the proposed scheme was used. The simulated UAV–Target environment is composed of 3D landmarks, which are randomly distributed over the ground. In this case, a UAV equipped with the required sensors is simulated. To include uncertainty into the simulations, the following Gaussian noise is added to measurements: for camera measurements  $\sigma_c = 4$  pixels; for altimeter measurements  $\sigma_a = 25$  cm; and for range sensor measurements  $\sigma_r = 25$  cm. All measurements are emulated to be acquired with a frequency of 10 Hz. The magnitude of the camera noise is bigger than the typical noise of real monocular measurement. In this way, it is intended to consider, in addition to the imperfection of the sensor, the errors in camera orientation due to the gimbal assumption. In simulations, the target was moved along a predefined trajectory.

#### 6.1.2. Convergence and Stability Tests

The objective of the test presented in this subsection is to show how the robustness of the SLAM system takes advantage of the inclusion of the altimeter measurements. In other words, both observability conditions described in Section 3 (with and without altimeter measurements) are tested. For this test, a control system is assumed to exist able to maintain the target tracking by the UAV.

It is well known that the initial conditions of the landmarks play an important role in the convergence and stability of SLAM systems. Therefore, a good way to evaluate the robustness of the SLAM system is to force bad initial conditions for the landmarks states. This means that (only for this test) the proposed initialization technique, described in Section 4.1, will not be used. Instead, the initial states of the landmarks  $\mathbf{x}_a^{new}$ , will be randomly determined using different standard deviations for the error position. Note that in this case, the initial conditions of  $\hat{\mathbf{x}}_t$ ,  $\hat{\mathbf{x}}_c$ ,  $\hat{\mathbf{v}}_t$ ,  $\hat{\mathbf{v}}_c$  are assumed to be exactly known.

Three different conditions of initial error are considered:  $\sigma_a = \{1, 2, 3\}$  meters, with a continuous uniform distribution. Figure 5 shows the actual trajectories followed by the target and the UAV.

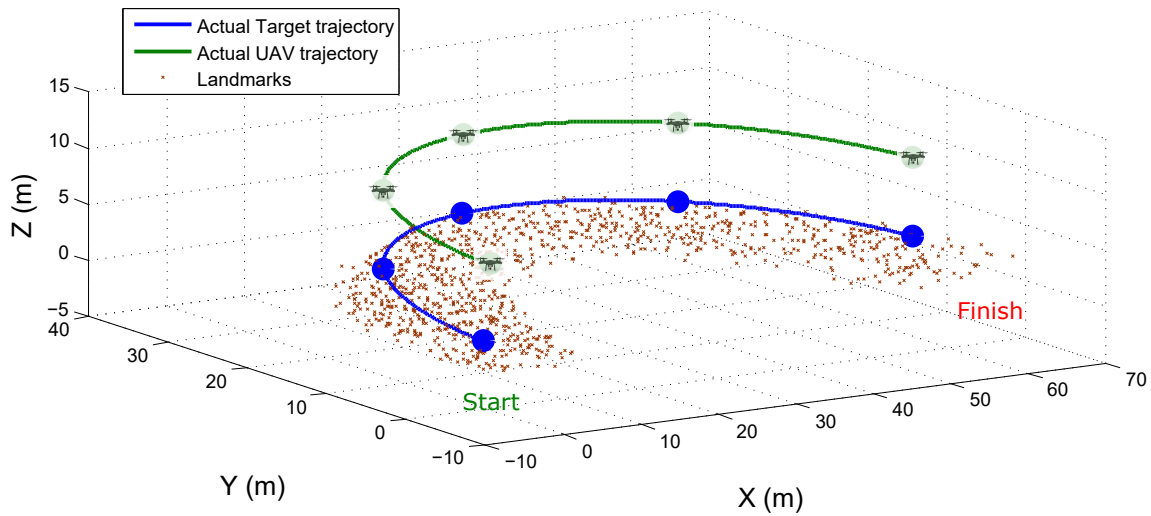


Figure 5. Actual UAV and target trajectories.

Figure 6 shows the results of the tests. The estimated positions of the UAV are plotted for each reference axis (row plots), and each column of plots shows the results obtained from each observability case. The results of the estimated state of the target are very similar to those presented for the UAV and, therefore, are omitted.

Table 2 summarizes the Mean Squared Error (MSE) of the estimated positions obtained for both the target and the UAV.

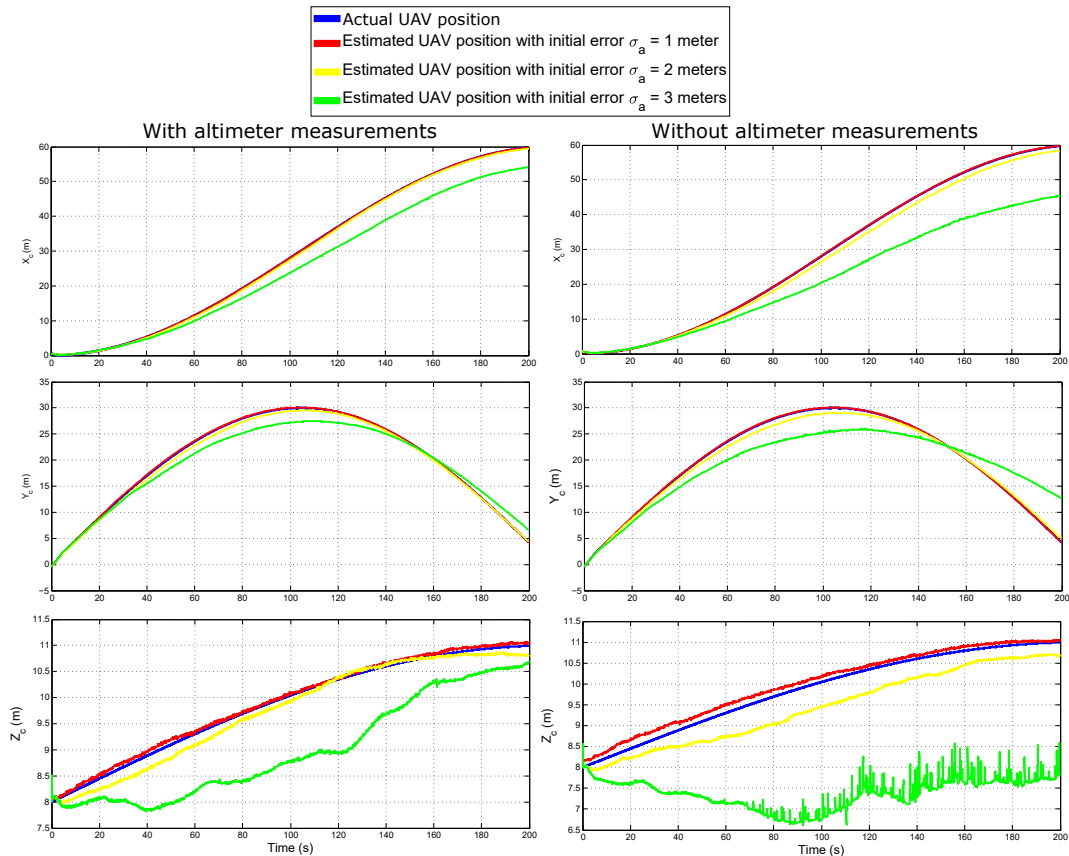


Figure 6. UAV estimated position.

**Table 2.** Mean Squared Error for the estimated position of the target ( $MSEX_t$ ,  $MSEY_t$ ,  $MSEZ_t$ ) and the estimated position of the UAV ( $MSEX_c$ ,  $MSEY_c$ ,  $MSEZ_c$ ).

		$MSEX_t$ (m)	$MSEY_t$ (m)	$MSEZ_t$ (m)	$MSEX_c$ (m)	$MSEY_c$ (m)	$MSEZ_c$ (m)
With Altimeter	$\sigma_a = 1$ m	0.0075	0.0187	0.0042	0.0045	0.0151	0.0033
	$\sigma_a = 2$ m	0.1214	0.2345	0.0302	0.1170	0.2309	0.0219
	$\sigma_a = 3$ m	18.9603	3.0829	0.9351	18.9578	3.0790	0.8962
Without Altimeter	$\sigma_a = 1$ m	0.0178	0.0139	0.0153	0.0145	0.0105	0.0207
	$\sigma_a = 2$ m	1.7179	0.4689	0.2379	1.7078	0.4686	0.2084
	$\sigma_a = 3$ m	80.9046	12.8259	7.3669	80.9000	12.8185	6.9981

Taking a closer look at Figure 6 and Table 2, it can be observed that both, the observability property and the initial conditions, play a preponderant role in the convergence and stability of the EKF-SLAM. For several applications, the initial position of the UAVs can be assumed to be known. However, in SLAM, the position of the map features must be estimated online. That confirms the great importance of using good features initialization techniques in visual-SLAM; and, as it can be expected, the better observability properties the better performance of the EKF-SLAM system, indeed.

### 6.1.3. Comparative Study

In this subsection a comparative study between the proposed monocular-based SLAM method and the following methods is presented,

- (1) Monocular SLAM
- (2) Monocular SLAM with anchors.
- (3) Monocular SLAM with inertial measurements.
- (4) Monocular SLAM with altimeter.
- (5) Monocular SLAM with a cooperative target: without target-based initialization.

There are some remarks about the methods used in the comparison. The method (1) is the approach described in [47]. This method is included only as a reference to highlight that purely monocular methods cannot retrieve the metric scale of the scene. The method (2) is similar to the previous method. But in this case, to set the metric scale of the estimates, the position of a subset of the landmarks seen in the first frame is assumed to be perfectly known (anchors). The method (3) is the approach described in [12]. In this case, inertial measurements obtained from an inertial measurement unit (IMU) are fused into the system. For this IMU-based method, it is assumed that the alignment of the camera and the inertial measurement unit is perfectly known; the dynamic error bias of the accelerometers is neglected as well. The method (4) is the approach proposed in [13]. In this case, altitude measurements given by an altimeter are fused into the monocular SLAM system. The method (5) is a variation of the proposed method. In this case, the landmark initialization technique proposed in Section 4.1 is not used, and instead only the regular initialization method is used. Therefore, this variation of the proposed method is included in the comparative study to highlight the advantage of the proposed cooperative-based initialization technique.

It is worth pointing out that all the methods (included the proposed method) use the same hypothetical initial depth for the landmarks without a priori inference of their position. Also for the comparative study, a control system is assumed to exist able to maintain the target tracking by the UAV.

#### First Comparison Test

Using the simulation setup illustrated in Figure 5, the performance of all the methods were tested for estimating the position of the camera-robot and the map of landmarks.

Figure 7 shows the results obtained from each method when estimating the position of the UAV. Figure 8 shows the results obtained from each method when estimating the velocity of the UAV. For the

sake of clarity, the results of Figures 7 and 8 are shown in two columns of plots. Each row of plots represents a reference axis.

Table 3 summarizes the Mean Squared Error (MSE) for the estimated relative position of the UAV expressed in each one of the three axes. Table 4 summarizes the Mean Squared Error (MSE) for the estimated position of the landmarks expressed in each one of the three axes.

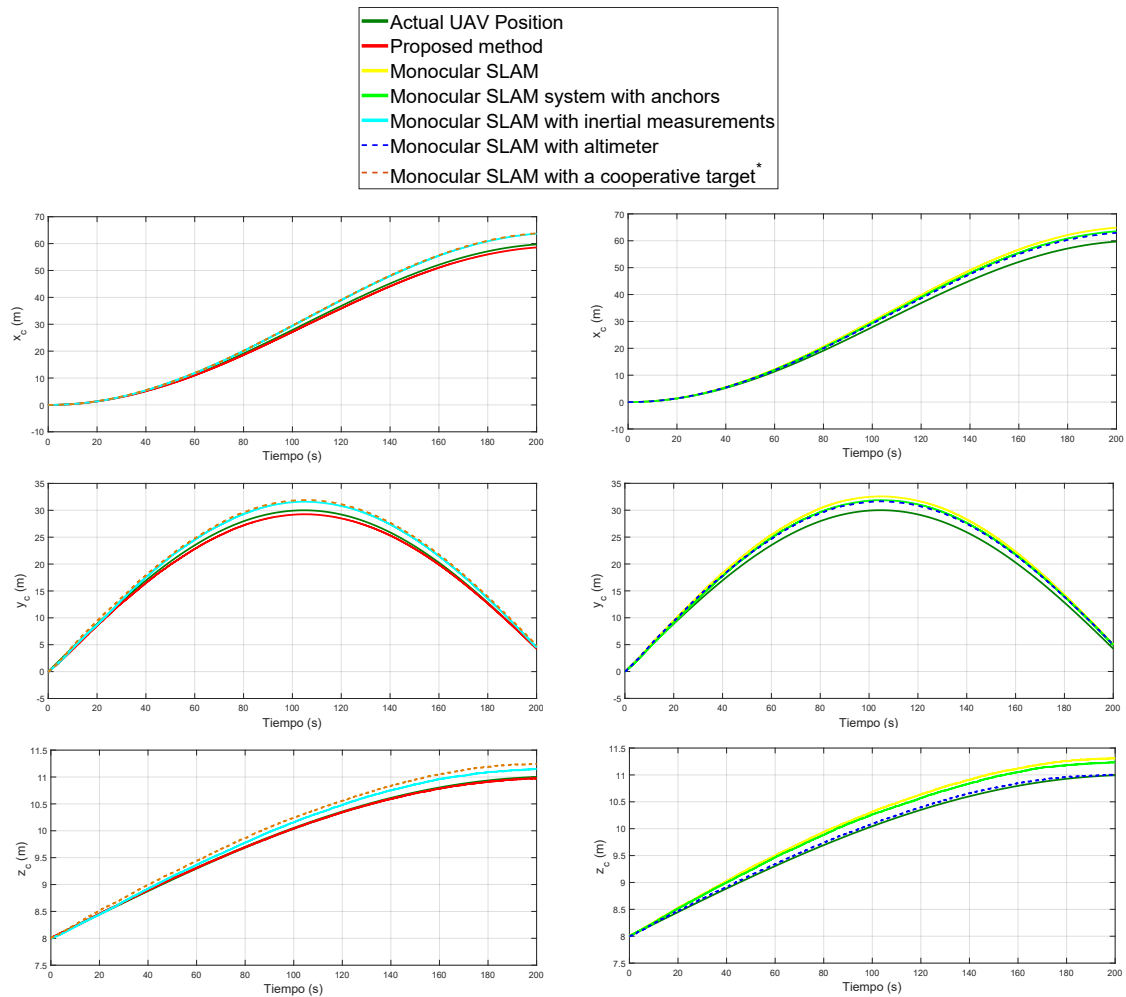


Figure 7. Comparison: UAV estimated position.

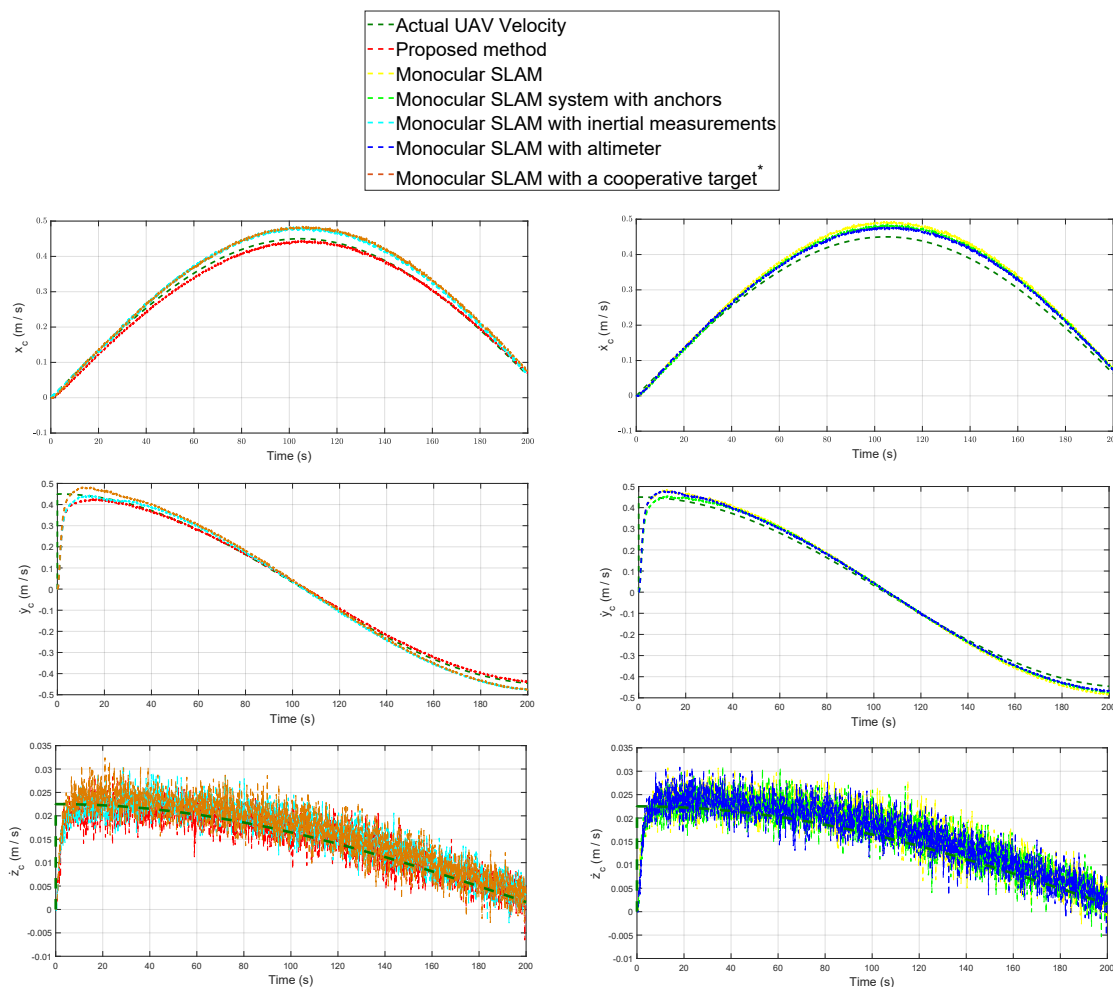


Figure 8. Comparison: UAV estimated velocity.

Table 3. Total Mean Squared Error for the estimated position of the UAV.

Method	MSEX (m)	MSEY (m)	MSEZ (m)
Proposed method	0.5848	0.2984	0.0001
Monocular SLAM	9.1325	3.6424	0.0642
Monocular SLAM with anchors	4.9821	1.8945	0.0394
Monocular SLAM with inertial measurements	4.9544	1.2569	0.0129
Monocular SLAM with altimeter	3.5645	1.5885	0.0016
Monocular SLAM with a cooperative target	5.5552	1.9708	0.0367

Table 4. Total Mean Squared Error for the estimated position of the landmarks.

Method	MSEX (m)	MSEY (m)	MSEZ (m)
Proposed method	0.6031	0.2926	0.1677
Monocular SLAM	8.1864	2.8295	0.3861
Monocular SLAM with anchors	4.4931	1.4989	0.2701
Monocular SLAM with inertial measurements	4.4739	0.9979	0.3093
Monocular SLAM with altimeter	3.2397	1.2609	0.3444
Monocular SLAM with a cooperative target	5.0374	1.5394	0.3054

Second Comparison Test

In this comparison test, the performance of all the methods was tested for recovering the metric scale of the estimates. For this test, the target and the UAV follow a circular trajectory for 30 s.

During the flight, the altitude of the UAV was changed (see Figure 9). In this case, it is assumed that all the landmarks on the map are seen from the first frame and that they are kept in the camera field of view throughout the simulation.

The scale factor  $s$  is given by [51]:

$$s = \frac{d_{real}}{d_{est}} \quad (49)$$

where  $d_{real}$  is the real distance, and  $d_{est}$  is the estimated distance. For the monocular SLAM problem, there exist different kind of distances and lots of data for real (and estimated) distances: distances between camera and landmarks, distances between landmarks, distances defined by the positions of the camera in time periods (camera trajectory), among other distances. Therefore, in practice, there is not such a standard convention for determining the metric scale. But in general, for determining the scale, the averages of multiple real and estimated distances are considered. In this work, authors propose to use the following approximation, which averages all the distances among the map features.

$$d_{real} = \frac{1}{\sum_{k=1}^{n-1} (n-k)} \sum_{i=1}^n \sum_{j=i+1}^n d^{ij} \quad d_{est} = \frac{1}{\sum_{k=1}^{n-1} (n-k)} \sum_{i=1}^n \sum_{j=i+1}^n \hat{d}^{ij} \quad (50)$$

Let  $d^{ij}$  represent the actual distance of the  $i$ -th landmark respect to the  $j$ -th landmark. Let  $\hat{d}^{ij}$  represent the estimated distance of the  $i$ -th landmark respect to the camera  $j$ -th landmark, and let  $n$  be the total number of landmarks. From (49), if the metric scale is perfectly recovered then  $s = 1$ .

For this test, an additional method has been added for comparison purposes. The *Monocular SLAM with altimeter (Loosely-coupled)* explicitly computes the metric scale by using the ratio between the altitude obtained from an altimeter, and the unscaled altitude obtained from a purely monocular SLAM system. The computed metric scale is used then for scaling the monocular SLAM estimates.

Case 1: The UAV follows a circular flight trajectory while varying its altitude (see Figure 9, upper plot). In this case, the UAV gets back to fly over its initial position, and thus, the initial landmarks are seen again (loop-closure).

Figure 9 (lower plot) shows the evolution of the metric scale obtained for each method. In this case, for each method, the metric scale converged to a value, and remains almost constant. Even the monocular SLAM method (yellow) which does not incorporate any metric information, and the monocular SLAM with anchors (green) that only includes metric information at the beginning of the trajectory, exhibit the same behavior. It is important to note that this is the expected behavior since the camera-robot is following a circular trajectory with loop closure where the initial low-uncertainty landmarks are revisited.

Case 2: The UAV follows the same flight trajectory illustrated in Figure 5. In this case, the UAV drifts apart from its initial position, and thus, the initial landmarks are never seen again.

Figure 10 (upper plot) shows the evolution of the metric scale obtained for each method. In this case, the monocular SLAM method (yellow) was manually tuned to have a good initial metric scale. The initial conditions of the other methods are alike as those of the *Case 1*, but the vertical limits of the plot have been adjusted for better visualization. Figure 10 (middle and lower plots respectively) shows the Euclidean mean error in position for the camera-robot  $e_c$  and the Euclidean mean error in position for the landmarks  $e_a$  for each method, where

$$e_c = \sqrt{(x_c - \hat{x}_c)^2 + (y_c - \hat{y}_c)^2 + (z_c - \hat{z}_c)^2}$$

$$e_a = \sqrt{\left(\frac{1}{n} \sum_{i=1}^n x_a^i - \hat{x}_a^i\right)^2 + \left(\frac{1}{n} \sum_{i=1}^n y_a^i - \hat{y}_a^i\right)^2 + \left(\frac{1}{n} \sum_{i=1}^n z_a^i - \hat{z}_a^i\right)^2}$$

Observing Figure 10, as could be expected, for the methods that continuously incorporate metric information into the system through additional sensors, the metric scale converges to a value, and remains approximately constant (after time > 100 s). On the other hand, the methods that do not continuously incorporate metric information (monocular SLAM and monocular SLAM with anchors), exhibit a drift in the metric scale. As one could also expect in SLAM without loop-closure, all the methods present some degree of drifting in position, both for the robot-camera trajectory and the landmarks. The above reasoning is independent of the drift in metric scale (the methods with low drift in scale also present drift in position). Evidently, it is convenient to maintain a low error/drift in scale, because it affects the error/drift in position.

It is interesting to note that the loosely-coupled method (purple) appears to be extremely sensitive to measurements noise. In this case, the increasing “noise” in error position is because the scale correction-ratio increases as the error in the scale of the purely monocular SLAM (yellow) also increases. In other terms, the signal-to-noise ratio (S/N) increases. Surely some adaptations can be done, as filtering the computed metric scale, but a trade-off would be introduced between the time of convergence and the reduction of noise effects.

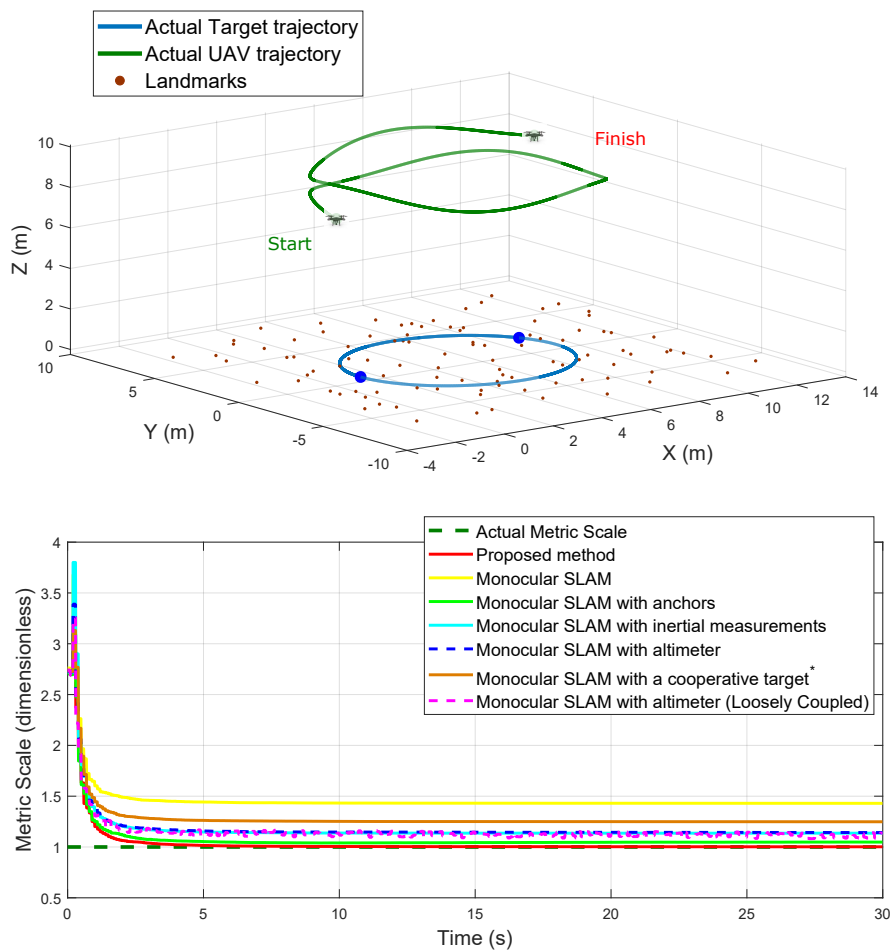
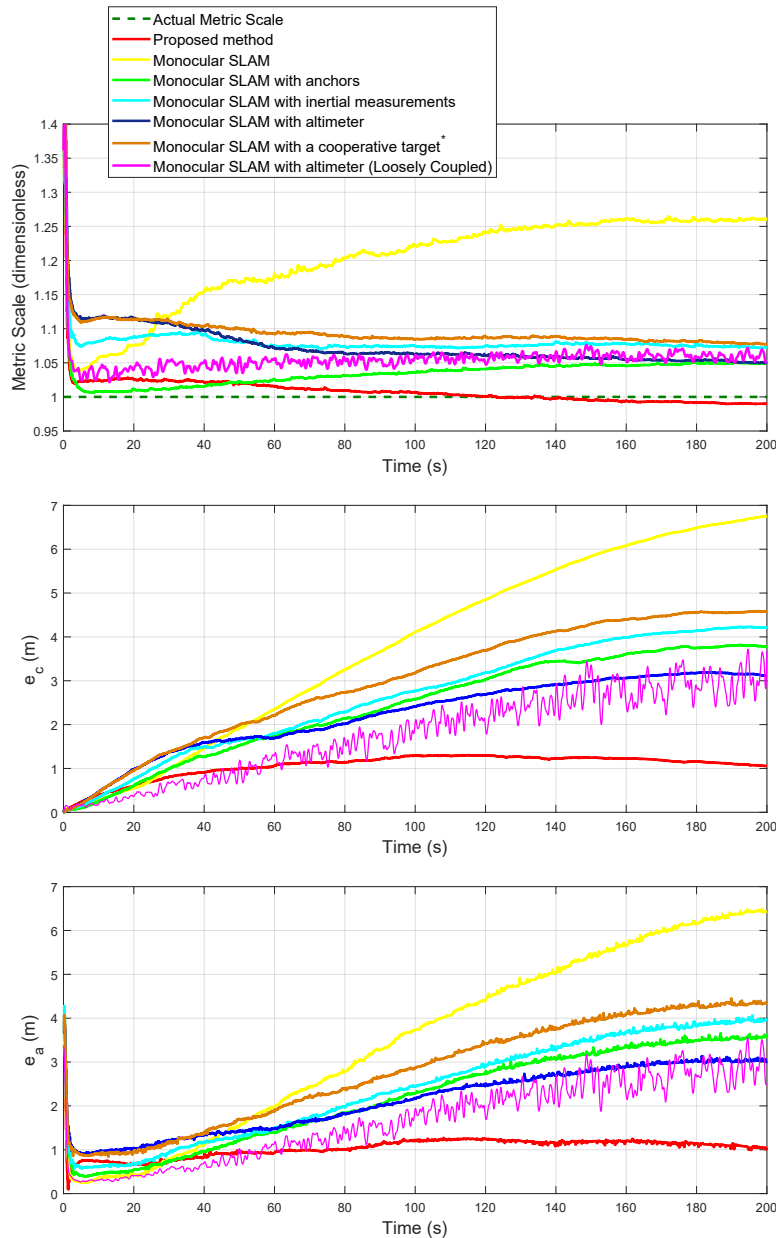


Figure 9. Case 1: Comparison of the estimated metric scale.





**Figure 10.** Case 2: Comparison of the estimated metric scale and Euclidean mean errors.

#### 6.1.4. Estimation and Control Test

A set of simulations were also carried out to test in a closed-loop manner the proposed control scheme. In this case, the estimates obtained from the proposed visual-based SLAM estimation method are used as feedback to the control scheme described in Section 5. The value of the vector  $\lambda_d$ , that defines the desired values of the servo visual and altitude control, is:  $\lambda_d = [0, 0, 7 + \sin(t \cdot 0.05), \text{atan2}(\hat{y}_q, \hat{x}_q)]^T$ . Those values for the desired control mean that the UAV has to remain flying exactly over the target at a varying relative altitude.

Figure 11 shows the evolution of the error respect to the desired values  $\lambda_d$ . In all the cases, note that the errors are bounded after an initial transient period.

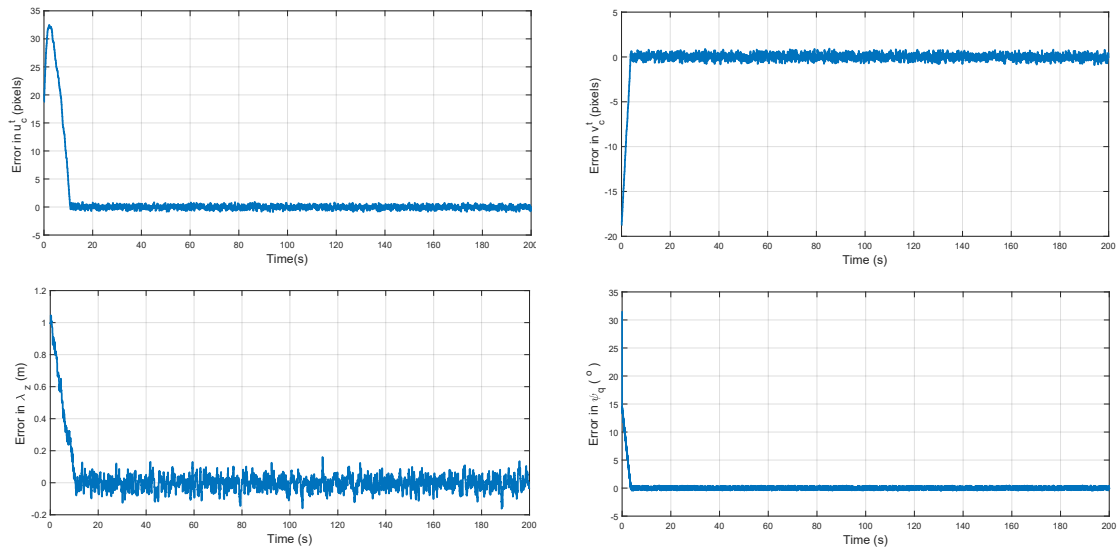


Figure 11. Errors with respect to  $\lambda_q$ .

Figure 12 shows the real and estimated position of the target and the UAV.

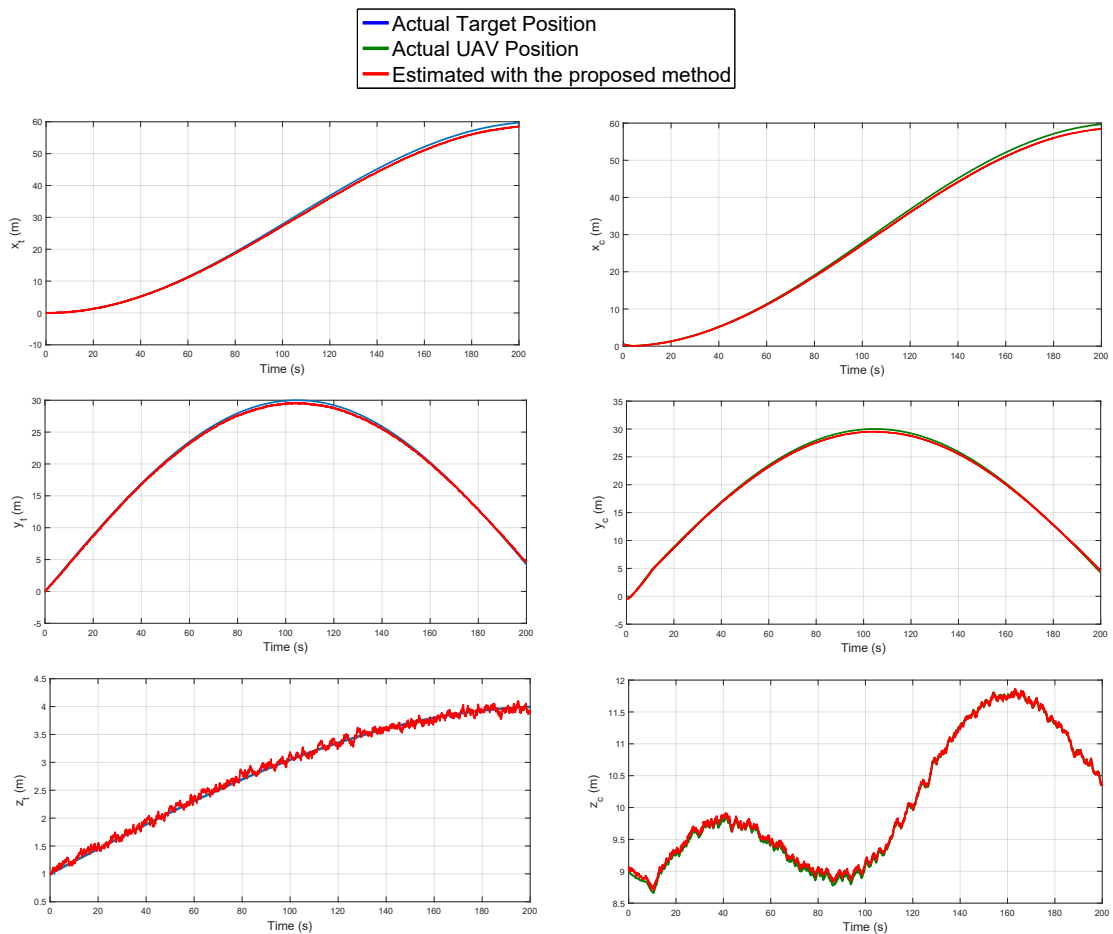


Figure 12. Estimated position of the target and the UAV obtained by the proposed method.

Table 5 summarizes the Mean Squared Error (MSE), expressed in each of the three axes, for the estimated position of: (i) the target, (ii) the UAV, and (iii) the landmarks.

**Table 5.** Mean Squared Error for the estimated position of target, UAV and landmarks.

	<b>MSEX (m)</b>	<b>MSEY (m)</b>	<b>MSEZ (m)</b>
Target	0.5149	0.0970	0.0036
UAV	0.5132	0.0956	0.0018
Landmarks	0.5654	0.1573	0.2901

Table 6 summarizes the Mean Squared Error (MSE) for the initial hypotheses of landmarks depth  $MSEd$ . Furthermore, Table 6 shows the Mean Squared Error (MSE) for the estimated position of landmarks, expressed in each of the three axes. In this case, since the landmarks near to the target are initialized with a small error, its final position is better estimated. Once again, this result shows the importance of the initialization process of landmarks in SLAM.

**Table 6.** Mean Squared Error for the the initial depth ( $MSEd$ ) and position estimation of the landmarks.

	<b>MSEd (m)</b>	<b>MSEX (m)</b>	<b>MSEY (m)</b>	<b>MSEZ (m)</b>
Far from the target	13.4009	3.5962	2.5144	7.6276
Near to the target	1.6216	0.5188	0.1280	1.6482

According to the above results, it can be seen that the proposed estimation method has a good performance to estimate the position of the UAV and the target. It can also be seen that the control system was able to maintain a stable flight formation along with all the trajectory respect to the target, using the proposed visual-based SLAM estimation system as a feedback.

### 6.2. Experiments with Real Data

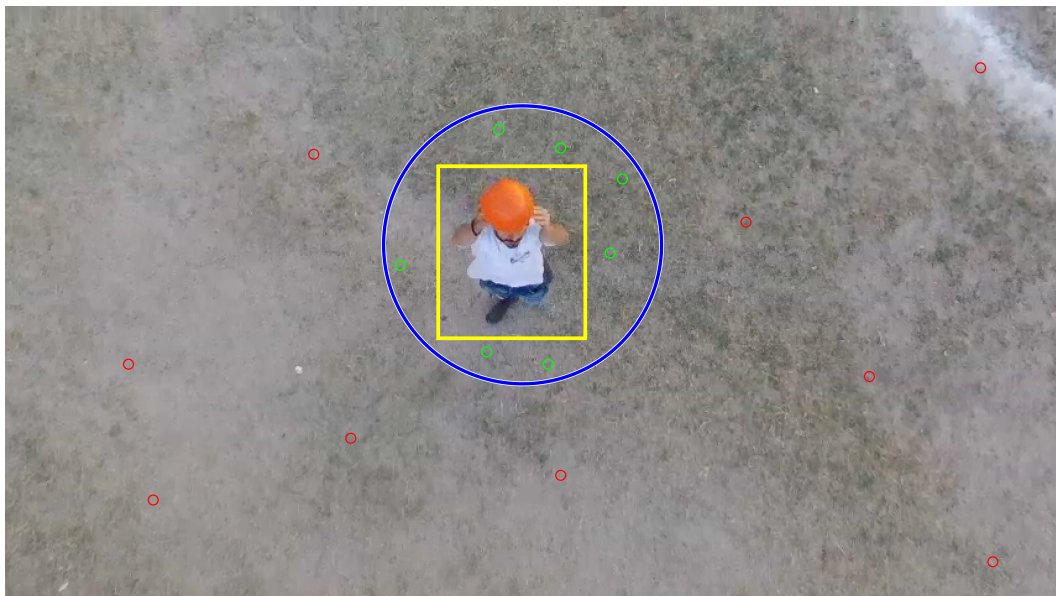
To test the proposed cooperative UAV–Target visual-SLAM method, an experiment with real data was carried out. In this case, a Parrot Bebop 2<sup>®</sup> quadcopter [33] (see Figure 13) was used for capturing real data with its sensory system.

**Figure 13.** Parrot Bebop 2<sup>®</sup> quadcopter.

The set of sensors of the Bebop 2 that were used in experiments consists of (i) a camera with a wide-angle lens and (ii) a barometer-based altimeter. The drone camera has a digital gimbal that allows to fulfill the assumption that the camera is always pointing to the ground. The vehicle was controlled through commands sent to it via Wi-Fi by a Matlab<sup>®</sup> application running in a ground-based PC. The same ground-based application was used for capturing, via Wi-Fi, the sensor data from the drone. In this case, camera frames with a resolution of  $856 \times 480$  pixels were captured at 24 fps. The altimeter signal was captured at 40 Hz. The range measurement between the UAV and the target was obtained by using the images and geometric information of the target. In experiments, the target was represented by a person walking with an orange ball over his head (See Figure 14). For evaluating the results

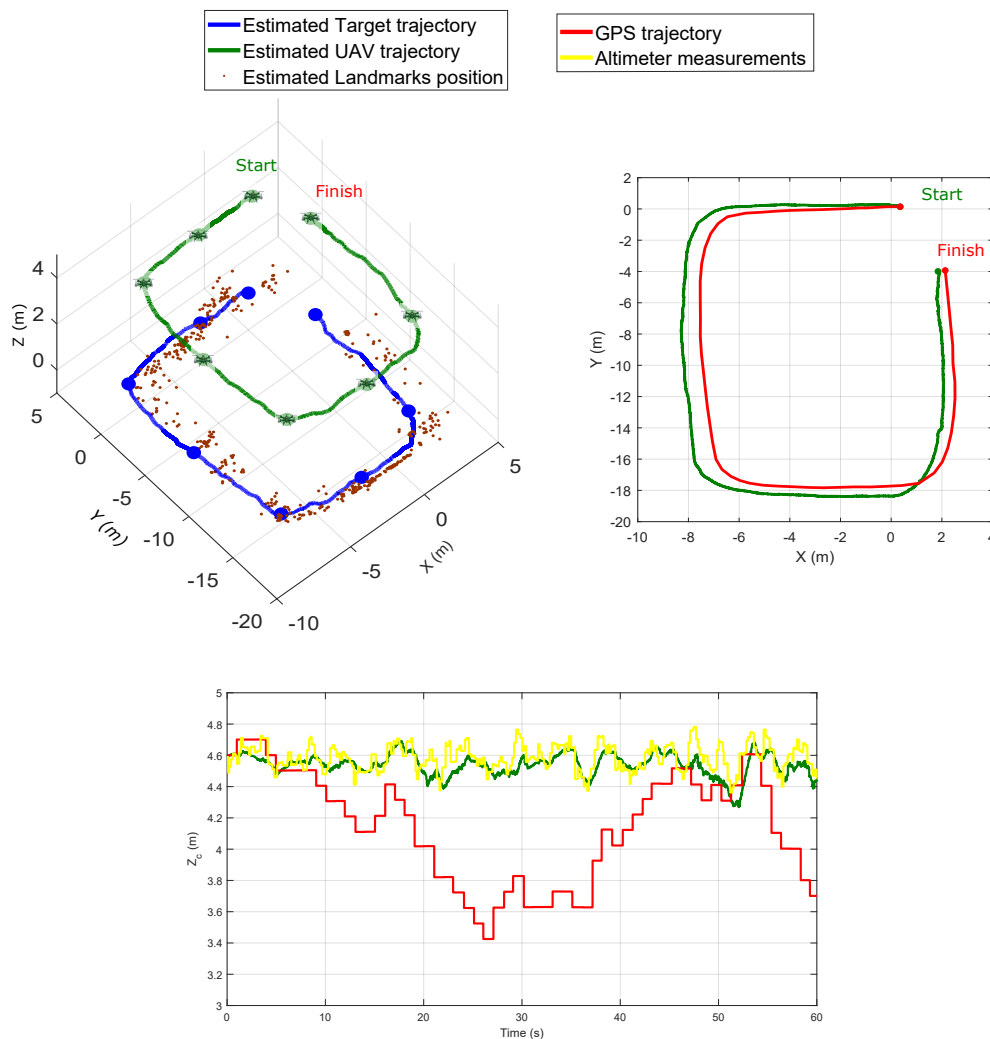
obtained with the proposed method, the on-board GPS device mounted on the quadcopter was used to obtain a flight trajectory reference. It is important to note that, due to the absence of an accurate ground truth, the relevance of the experiment is two-fold: (i) to show that the proposed method can be practically implemented with commercial hardware; and (ii) to demonstrate that using only the main camera and the altimeter of Bebop 2, the proposed method can provide similar navigation capabilities than the original Bebop's navigation system (which additionally integrate GPS, ultrasonic sensor, and optical flow sensor), in scenarios where a cooperative target is available.

Figure 14 shows a frame taken by the UAV on-board camera. The detection of the target is highlighted with a yellow bounding box. The search area of landmarks near the target is highlighted with a blue circle centered on the target. For the experiment, a radius of 1 m was chosen for the sphere centered on the target that is used for discriminating the landmarks. In this frame, some visual characteristics are detected in the image. The red circles indicate those visual features that are not within the search area near the target, that is, inside the blue circle. Instead, the green circles indicate those detected features within the search area. The visual features that are found within the patch that corresponds to the target (yellow box) are neglected, this behaviour is to avoid considering any visual feature that belongs to the target as a static landmark of the environment.



**Figure 14.** Frame captured by the UAV on-board camera.

Figure 15 shows both the UAV and the target estimated trajectories. This figure also shows the trajectory of the UAV given by the GPS and the altitude measurements supplied by the altimeter. Although the trajectory given by the GPS cannot be considered as a perfect ground-truth (especially for the altitude), it is still useful as a reference for evaluating the performance of the proposed visual-based SLAM method, and most especially if the proposed method is intended to be used in scenarios where the GPS is not available or reliable enough. According to the experiments with real data, it can be appreciated that the UAV trajectory has been estimated fairly well.



**Figure 15.** Comparison between the trajectory estimated with the proposed method, the GPS trajectory and the altitude measurements.

## 7. Conclusions

This work presented a cooperative visual-based SLAM system that allows an aerial robot following a cooperative target to estimate the states of the robot as well as the target in GPS-denied environments. This objective has been achieved using monocular measurements of the target and the landmarks, measurements of altitude of the UAV, and range measurements between UAV and target.

The observability property of the system was investigated by carrying out a nonlinear observability analysis. In this case, a contribution has been to show that the inclusion of altitude measurements improves the observability properties of the system. Furthermore, a novel technique to estimate the approximate depth of the new visual landmarks was proposed.

In addition to the proposed estimation system, a control scheme was proposed, allowing to control the flight formation of the UAV with respect to the cooperative target. The stability of control laws has been proven using the Lyapunov theory.

An extensive set of computer simulations and experiments with real data were performed to validate the theoretical findings. According to the simulations and experiments with real data results, the proposed system has shown a good performance to estimate the position of the UAV and the target. Moreover, with the proposed control laws, the proposed SLAM system shows a good closed-loop performance.

**Author Contributions:** Conceptualization, R.M. and A.G.; methodology, S.U. and R.M.; software, J.-C.T. and E.G.; validation, J.-C.T., S.U. and E.G.; investigation, S.U. and R.M.; resources, J.-C.T. and E.G.; writing—original draft preparation, J.-C.T. and R.M.; writing—review and editing, R.M. and A.G.; supervision, R.M. and A.G.; funding acquisition, A.G. All authors have read and agreed to the published version of the manuscript.

**Acknowledgments:** This research has been funded by Project DPI2016-78957-R, Spanish Ministry of Economy, Industry and Competitiveness.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Lie Derivatives of Measurements

In this appendix, the Lie derivatives for each measurement equation used in Section 3, are presented.

From Equations (3) and (1), the zero-order Lie derivative can be obtained for landmark projection model:

$$\frac{\partial(\mathcal{L}_f^0(\mathbf{h}_c^i))}{\partial \mathbf{x}} = \left[ \mathbf{0}_{2 \times 6} \mid -\mathbf{H}_c^i \mathbf{W} \mathbf{R}_c \mathbf{0}_{2 \times 3} \mid \mathbf{0}_{2 \times 3(i-1)} \mathbf{H}_c^i \mathbf{W} \mathbf{R}_c \mathbf{0}_{2 \times 3(n_1-i)} \right] \quad (\text{A1})$$

where

$$\mathbf{H}_c^i = \frac{f_c}{(z_d^i)^2} \begin{bmatrix} \frac{z_d^i}{d_u} & 0 & -\frac{x_d^i}{d_u} \\ 0 & \frac{z_d^i}{d_v} & -\frac{y_d^i}{d_v} \end{bmatrix} \quad (\text{A2})$$

The first-order Lie Derivative for landmark projection model is:

$$\frac{\partial(\mathcal{L}_f^1(\mathbf{h}_c^i))}{\partial \mathbf{x}} = \left[ \mathbf{0}_{2 \times 6} \mid \mathbf{H}_{dc}^i \quad -\mathbf{H}_c^i \mathbf{W} \mathbf{R}_c \mid \mathbf{0}_{2 \times 3(i-1)} \quad -\mathbf{H}_{dc}^i \quad \mathbf{0}_{2 \times 3(n_1-i)} \right] \quad (\text{A3})$$

where

$$\mathbf{H}_{dc}^i = \left[ \mathbf{H}_1^i \quad \mathbf{H}_2^i \quad \mathbf{H}_3^i \right] \left( \mathbf{W} \mathbf{R}_c \right)^2 \mathbf{v}_c \quad (\text{A4})$$

and

$$\mathbf{H}_1^i = \frac{f_c}{d_u (z_d^i)^2} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{H}_2^i = \frac{f_c}{d_v (z_d^i)^2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad \mathbf{H}_3^i = \frac{f_c}{(z_d^i)^3} \begin{bmatrix} -\frac{z_d^i}{d_u} & 0 & \frac{2x_d^i}{d_u} \\ 0 & -\frac{z_d^i}{d_v} & \frac{2y_d^i}{d_v} \end{bmatrix} \quad (\text{A5})$$

From Equations (5) and (1), the zero-order Lie derivative can be obtained for target projection model:

$$\frac{\partial(\mathcal{L}_f^0(\mathbf{h}_c^t))}{\partial \mathbf{x}} = \left[ \mathbf{H}_c^t \mathbf{W} \mathbf{R}_c \mathbf{0}_{2 \times 3} \mid -\mathbf{H}_c^t \mathbf{W} \mathbf{R}_c \mathbf{0}_{2 \times 3} \mid \mathbf{0}_{2 \times 3n_1} \right] \quad (\text{A6})$$

where

$$\mathbf{H}_c^t = \frac{f_c}{(z_d^t)^2} \begin{bmatrix} \frac{z_d^t}{d_u} & 0 & -\frac{x_d^t}{d_u} \\ 0 & \frac{z_d^t}{d_v} & -\frac{y_d^t}{d_v} \end{bmatrix} \quad (\text{A7})$$

The first-order Lie Derivative for target projection model is:

$$\frac{\partial(\mathcal{L}_f^1(\mathbf{h}_c^t))}{\partial \mathbf{x}} = \left[ -\mathbf{H}_{dc}^t \quad \mathbf{H}_c^t \mathbf{W} \mathbf{R}_c \mid \mathbf{H}_{dc}^t \quad -\mathbf{H}_c^t \mathbf{W} \mathbf{R}_c \mid \mathbf{0}_{2 \times 3n_1} \right] \quad (\text{A8})$$

where

$$\mathbf{H}_{dc}^t = \left[ \mathbf{H}_1^t \quad \mathbf{H}_2^t \quad \mathbf{H}_3^t \right] \left( \mathbf{W} \mathbf{R}_c \right)^2 (\mathbf{v}_c - \mathbf{v}_t) \quad (\text{A9})$$

and

$$\mathbf{H}_1^t = \frac{f_c}{d_u(z_d^t)^2} \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{H}_2^t = \frac{f_c}{d_v(z_d^t)^2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad \mathbf{H}_3^t = \frac{f_c}{(z_d^t)^3} \begin{bmatrix} -\frac{z_d^t}{d_u} & 0 & \frac{2x_d^t}{d_u} \\ 0 & -\frac{z_d^t}{d_v} & \frac{2y_d^t}{d_v} \end{bmatrix} \quad (\text{A10})$$

From Equations (7) and (1), the zero-order Lie derivative can be obtained for the altimeter measurement model:

$$\frac{\partial(\mathcal{L}_f^0(h_a))}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_{1 \times 6} & | & \mathbf{0}_{1 \times 2} & 1 & \mathbf{0}_{1 \times 3} & | & \mathbf{0}_{1 \times 3n_1} \end{bmatrix} \quad (\text{A11})$$

The first-order Lie Derivative for the altimeter measurement model is:

$$\frac{\partial(\mathcal{L}_f^1(h_a))}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0}_{1 \times 6} & | & \mathbf{0}_{1 \times 5} & 1 & | & \mathbf{0}_{1 \times 3n_1} \end{bmatrix} \quad (\text{A12})$$

From Equations (8) and (1), the zero-order Lie derivative can be obtained for the range sensor model:

$$\frac{\partial(\mathcal{L}_f^0(h_r))}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{H}_r & \mathbf{0}_{1 \times 3} & | & -\mathbf{H}_r & \mathbf{0}_{1 \times 3} & | & \mathbf{0}_{1 \times 3n_1} \end{bmatrix} \quad (\text{A13})$$

where

$$\mathbf{H}_r = \begin{bmatrix} \frac{x_t - x_c}{h_r} & \frac{y_t - y_c}{h_r} & \frac{z_t - z_c}{h_r} \end{bmatrix} \quad (\text{A14})$$

The first-order Lie Derivative for the range sensor model is:

$$\frac{\partial(\mathcal{L}_f^1(h_r))}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{H}_{dr} & \mathbf{H}_r & | & -\mathbf{H}_{dr} & -\mathbf{H}_r & | & \mathbf{0}_{1 \times 3n_1} \end{bmatrix} \quad (\text{A15})$$

where

$$(\mathbf{H}_{dr})^T = \frac{1}{h_r} \left[ \mathbf{I}_3 - (\mathbf{H}_r)^T \mathbf{H}_r \right] (\mathbf{v}_t - \mathbf{v}_c) \quad (\text{A16})$$

## Appendix B. Proof of the Existence of $\hat{\mathbf{B}}^{-1}$

In this appendix, the proof of the existence of  $\hat{\mathbf{B}}^{-1}$  is presented. For this purpose, it is necessary to demonstrate that  $|\hat{\mathbf{B}}| \neq 0$ . From Equation (40),  $|\hat{\mathbf{B}}| = |\hat{\mathbf{M}} \hat{\mathbf{\Omega}}|$ , where

$$\hat{\mathbf{M}} = - \begin{bmatrix} \hat{\mathbf{J}}_c^t \mathbf{W} \mathbf{R}_c & \mathbf{0}_{2 \times 1} \\ \mathbf{c}_1 & \mathbf{c}_2 \end{bmatrix} \quad (\text{A17})$$

using  $|\hat{\mathbf{B}}| = |\hat{\mathbf{M}} \hat{\mathbf{\Omega}}| = |\hat{\mathbf{M}}| |\hat{\mathbf{\Omega}}|$ . From Equation (41), then  $|\hat{\mathbf{\Omega}}| = 1$ . For this work, given the assumptions for matrix  $\mathbf{W} \mathbf{R}_c$  (see Section 2), the following expression is defined:

$$\mathbf{W} \mathbf{R}_c = - \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (\text{A18})$$

based on the previous expressions, then  $|\hat{\mathbf{M}}| = -\frac{(f_c)^2}{(z_d^t)^2 d_u d_v}$ . Finally,  $|\hat{\mathbf{B}}| = -\frac{(f_c)^2}{(z_d^t)^2 d_u d_v}$ . Since  $f_c, d_u, d_v, z_d^t > 0$ , then,  $|\hat{\mathbf{B}}| \neq 0$ , therefore  $\hat{\mathbf{B}}^{-1}$  exists.



## References

1. Xu, Z.; Douillard, B.; Morton, P.; Vlaskine, V. Towards Collaborative Multi-MAV-UGV Teams for Target Tracking. In Proceedings of the 2012 Robotics: Science and Systems Workshop Integration of Perception with Control and Navigation for Resource-Limited, Highly Dynamic, Autonomous Systems, 9–12 July 2012, Sydney, Australia.
2. Michael, N.; Shen, S.; Mohta, K. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *J. Field Robot.* **2012**, *29*, 832–841. [[CrossRef](#)]
3. Hu, H.; Wei, N. A study of GPS jamming and anti-jamming. In Proceedings of the 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS), Shenzhen, China, 19–20 December 2009; Volume 1, pp. 388–391.
4. Bachrach, S.; Prentice, R.H.; Roy, N. RANGE-Robust autonomous navigation in GPS-denied environments. *J. Field Robot.* **2011**, *5*, 644–666. [[CrossRef](#)]
5. Meguro, J.I.; Murata, T.; Takiguchi, J.I.; Amano, Y.; Hashizume, T. GPS multipath mitigation for urban area using omnidirectional infrared camera. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 22–30. [[CrossRef](#)]
6. Davison, A.; Reid, I.; Molton, N.; Stasse, O. Monoslam: Realtime single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [[CrossRef](#)] [[PubMed](#)]
7. Artieda, J.; Sebastian, J.M.; Campoy, P.; Correa, J.F.; Mondragón, I.F.; Martínez, C.; Olivares, M. Visual 3-d slam from uavs. *J. Intell. Robot. Syst.* **2009**, *55*, 299–321. [[CrossRef](#)]
8. Weiss, S.; Scaramuzza, D.; Siegwart, R. Monocular-slam based navigation for autonomous micro helicopters in gps-denied environments. *J. Field Robot.* **2011**, *28*, 854–874. [[CrossRef](#)]
9. Mirzaei, F.; Roumeliotis, S. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *IEEE Trans. Robot.* **2008**, *24*, 1143–1156. [[CrossRef](#)]
10. Celik, K.; Somani, A.K. Monocular vision slam for indoor aerial vehicles. *J. Electr. Comput. Eng.* **2013**, *2013*, 374165.
11. Nutzi, G.; Weiss, S.; Scaramuzza, D.; Siegwart, R. Fusion of imu and vision for absolute scale estimation in monocular slam. *J. Intell. Robot. Syst.* **2011**, *61*, 287–299. [[CrossRef](#)]
12. Wang, C.L.; Wang, T.M.; Liang, J.H.; Zhang, Y.C.; Zhou, Y. Bearing-only visual slam for small unmanned aerial vehicles in gps-denied environments. *Int. J. Autom. Comput.* **2014**, *10*, 387–396. [[CrossRef](#)]
13. Urzua, S.; Munguía, R.; Nuño, E.; Grau, A. Minimalistic approach for monocular SLAM system applied to micro aerial vehicles in GPS-denied environments. *Trans. Inst. Meas. Control.* **2018**. [[CrossRef](#)]
14. Mourikis, A.I.; Roumeliotis, S.I. Performance Bounds for Cooperative Simultaneous Localisation and Mapping (C-SLAM). In Proceedings of the Robotics: Science and Systems Conference, Cambridge, MA, USA, 8–11 June 2005.
15. Fenwick, J.W.; Newman, P.M.; Leonard, J.J. Cooperative Concurrent Mapping and Localisation. In Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 11–15 May 2002.
16. Chowdhary, G.; Johnson, E.N.; Magree, D.; Wu, A.; Shein, A. GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft. *J. Field Robot.* **2013**, *30*, 415–438. [[CrossRef](#)]
17. Vetrella, A.R.; Opromolla, R.; Fasano, G.; Accardo, D.; Grassi, M. Autonomous Flight in GPS-Challenging Environments Exploiting Multi-UAV Cooperation and Vision-aided Navigation. In Proceedings of the AIAA Information Systems, Grapevine, TX, USA, 10–14 July 2017.
18. Vetrella, A.R.; Fasano, G.; Accardo, D. Cooperative Navigation in GPS-Challenging Environments Exploiting Position Broadcast and Vision-based Tracking. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems, Arlington, VA, USA, 7–10 June 2016.
19. Guerra, E.; Munguia, R.; Grau, A. Monocular SLAM for Autonomous Robots with Enhanced Features Initialization. *Sensors* **2014**, *14*, 6317–6337. [[CrossRef](#)] [[PubMed](#)]
20. Ding, S.; Liu, G.; Li, Y.; Zhang, J.; Yuan, J.; Sun, F. SLAM and Moving Target Tracking Based on Constrained Local Submap Filter. In Proceedings of the 2015 IEEE International Conference on Information and Automation, Lijiang, China, 8–10 August 2015.
21. Ahmad, A.; Tipaldi, G.D.; Lima, P.; Burgard, W. Cooperative robot localization and target tracking based on least squares minimization. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013.

22. Han, Y.; Wei, C.; Li, R.; Wang, J.; Yu, H. A Novel Cooperative Localization Method Based on IMU and UWB. *Sensors* **2020**, *20*, 467. [[CrossRef](#)]
23. Molina Martel, F.; Sidorenko, J.; Bodensteiner, C.; Arens, M.; Hugentobler, U. Unique 4-DOF Relative Pose Estimation with Six Distances for UWB/V-SLAM-Based Devices. *Sensors* **2019**, *19*, 4366. [[CrossRef](#)]
24. Trujillo, J.C.; Munguia, R.; Guerra, E.; Grau, A. Visual-Based SLAM Configurations for Cooperative Multi-UAV Systems with a Lead Agent: An Observability-Based Approach. *Sensors* **2018**, *18*, 4243. [[CrossRef](#)]
25. Jin, Q.; Liu, Y.; Li, F. Visual SLAM with RGB-D Cameras. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 4072–4077.
26. Sun, F.; Sun, X.; Guan, B.; Li, T.; Sun, C.; Liu, Y. Planar Homography Based Monocular SLAM Initialization Method. In Proceedings of the 2019 2nd International Conference on Service Robotics Technologies, Beijing, China, 22–24 March 2019; pp. 48–52.
27. Zhang, Z.; Zhao, R.; Liu, E.; Yan, K.; Ma, Y. Scale Estimation and Correction of the Monocular Simultaneous Localization and Mapping (SLAM) Based on Fusion of 1D Laser Range Finder and Vision Data. *Sensors* **2018**, *18*, 1948. [[CrossRef](#)]
28. Reif, K.; Günther, S.; Yaz, E.; Unbehauen, R. Stochastic stability of the discrete-time extended Kalman filter. *IEEE Trans. Autom. Control* **1999**, *44*, 714–728. [[CrossRef](#)]
29. Kluge, S.; Reif, K.; Brokate, M. Stochastic stability of the extended Kalman filter with intermittent observations. *IEEE Trans. Autom. Control* **2010**, *55*, 514–518. [[CrossRef](#)]
30. Munguía, R.; Grau, A. Concurrent Initialization for Bearing-Only SLAM. *Sensors* **2010**, *10*, 1511–1534. [[CrossRef](#)]
31. Euston, M.; Coote, P.; Mahony, R.; Kim, J.; Hamel, T. A complementary filter for attitude estimation of a fixed-wing UAV. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 340–345.
32. Munguia, R.; Grau, A. A Practical Method for Implementing an Attitude and Heading Reference System. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 62. [[CrossRef](#)]
33. Parrot Bebop 2 Drone User Manual. Available online: <https://www.parrot.com/us/user-guide-bebop-2-fpv-us> (accessed on 21 June 2020).
34. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, MA, USA, 2003.
35. Srisamosorn, V.; Kuwahara, N.; Yamashita, A.; Ogata, T. Human-tracking System Using Quadrotors and Multiple Environmental Cameras for Face-tracking Application. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1729881417727357. [[CrossRef](#)]
36. Benezeth, Y.; Emile, B.; Laurent, H.; Rosenberger, C. Vision-Based System for Human Detection and Tracking in Indoor Environment. *Int. J. Soc. Robot.* **2010**, *2*, 41–52. [[CrossRef](#)]
37. Olivares-Mendez, M.A.; Fu, C.; Ludvig, P.; Bissyandé, T.F.; Kannan, S.; Zurad, M.; Annaiyan, A.; Voos, H.; Campoy, P. Towards an Autonomous Vision-Based Unmanned Aerial System against Wildlife Poachers. *Sensors* **2015**, *15*, 31362–31391. [[CrossRef](#)]
38. Briese, C.; Seel, A.; Andert, F. Vision-based detection of non-cooperative UAVs using frame differencing and temporal filter. In Proceedings of the International Conference on Unmanned Aircraft Systems, Dallas, TX, USA, 12–15 June 2018.
39. Mejías, L.; McNamara, S.; Lai, J. Vision-based detection and tracking of aerial targets for UAV collision avoidance. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 87–92.
40. Beard, R.W.; McLain, T.W. *Small Unmanned Aircraft: Theory and Practice*; Princeton University Press: Princeton, NJ, USA, 2012.
41. Alavi, B.; Pahlavan, K. Modeling of the TOA-based distance measurement error using UWB indoor radio measurements. *IEEE Commun. Lett.* **2006**, *10*, 275–277. [[CrossRef](#)]
42. Lanzisera, S.; Zats, D.; Pister, K.S.J. Radio frequency time-of-flight distance measurement for low-cost wireless sensor localization. *IEEE Sens. J.* **2011**, *11*, 837–845. [[CrossRef](#)]
43. Hermann, R.; Krener, A. Nonlinear controllability and observability. *IEEE Trans. Autom. Control* **1977**, *22*, 728–740. [[CrossRef](#)]
44. Slotine, J.E.; Li, W. *Applied Nonlinear Control*; Prentice-Hall Englewood Cliffs: Upper Saddle River, NJ, USA, 1991.

45. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part i. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [[CrossRef](#)]
46. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (slam): Part ii. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117. [[CrossRef](#)]
47. Montiel, J.M.M.; Civera, J.; Davison, A. Unified inverse depth parametrization for monocular SLAM. In Proceedings of the Robotics: Science and Systems Conference, Philadelphia, PA, USA, 16–19 August 2006.
48. Vega, L.L.; Toledo, B.C.; Loukianov, A.G. Robust block second order sliding mode control for a quadrotor. *J. Frankl. Inst.* **2012**, *349*, 719–739. [[CrossRef](#)]
49. Emran, B.J.; Yesildirek, A. Robust Nonlinear Composite Adaptive Control of Quadrotor. *Int. J. Digit. Inf. Wirel. Commun.* **2014**, *4*, 45–57. [[CrossRef](#)]
50. Utkin, V.I. Sliding Mode Control Design Principles and Applications to Electric Drives. *IEEE Trans. Ind. Electron.* **1993**, *40*, 23–36. [[CrossRef](#)]
51. Hanel, A.; Mitschke, A.; Boerner, R.; Van Opdenbosch, D.; Brodie, D.; Stilla, U. Metric Scale Calculation For Visual Mapping Algorithms. In Proceedings of the ISPRS Technical Commission II Symposium 2018, Riva del Garda, Italy, 3–7 June 2018.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).