

Article

Scale Factor Estimation for Quadrotor Monocular-Vision Positioning Algorithms

Alejandro Gómez-Casasola ^{*,†} and Hugo Rodríguez-Cortés [†] 

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional,
Av. Instituto Politécnico Nacional 2508, Col. San Pedro Zacatenco, Ciudad de Mexico 07360, Mexico

* Correspondence: angel.gomez@cinvestav.mx

† These authors contributed equally to this work.

Abstract: Unmanned aerial vehicle (UAV) autonomous navigation requires access to translational and rotational positions and velocities. Since there is no single sensor to measure all UAV states, it is necessary to fuse information from multiple sensors. This paper proposes a deterministic estimator to reconstruct the scale factor of the position determined by a simultaneous localization and mapping (SLAM) algorithm onboard a quadrotor UAV. The position scale factor is unknown when the SLAM algorithm relies on the information from a monocular camera. Only onboard sensor measurements can feed the estimator; thus, a deterministic observer is designed to rebuild the quadrotor translational velocity. The estimator and the observer are designed following the immersion and invariance method and use inertial and visual measurements. Lyapunov's arguments prove the asymptotic convergence of observer and estimator errors to zero. The proposed estimator's and observer's performance is validated through numerical simulations using a physics-based simulator.

Keywords: sensor fusion; quadrotor; observer design



Citation: Gómez-Casasola, A.; Rodríguez-Cortés, H. Scale Factor Estimation for Quadrotor Monocular-Vision Positioning Algorithms. *Sensors* **2022**, *22*, 8048. <https://doi.org/10.3390/s22208048>

Academic Editors: Chris Rizos, Allison Kealy, Jacek Paziewski and Yang Gao

Received: 12 September 2022

Accepted: 1 October 2022

Published: 21 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, quadrotors are used in various applications, thanks to their low cost, mechanical robustness, and high maneuverability. Such applications include homeland security, forest-fire control, surveillance, sea and land exploration, human search and rescue, archaeological exploration, and volcanic activity monitoring, among many others [1,2]. Most of the abovementioned applications become impractical or even dangerous for human operators; thus, autonomous navigation, control, and guidance are required.

A quadrotor can perform autonomous navigation in unknown environments when its autopilot has access to all states. However, providing access to all quadrotor states without relying on a remote computer or sensors demands a vehicle with the capacity to process and extract information from only onboard sensors. The bottleneck to measuring all quadrotor states is that there are no out-of-the-box functional and reliable sensors to measure all states directly. For example, the quadrotor's attitude is obtained by fusing the measurements from an inertial measurement unit (IMU). On the other hand, the global position system (GPS) is the primary sensor used for quadrotor positioning, but it presents limitations. It fails in environments where satellite communication is degraded, called GPS-denied environments, such as water bodies and indoors [3]. Furthermore, low-cost GPS does not provide enough resolution for trajectories on a centimeter scale, and the price of a GPS with higher resolution, such as differential GPS, increases drastically.

An algorithm to fuse GPS measurements with optical flow information using a Kalman filter (KF) was proposed in [4]. Shortly after, the work reported in [5] presented an improved sensor fusion algorithm based on an extended Kalman filter (EKF) that includes the measurements from an inertial navigation system (INS). Both sensor fusion algorithms improved the position estimation for low-cost GPS but not for GPS-denied environments.

Computer vision has emerged as a powerful solution for quadrotor position estimation. Visual sensors have many advantages over other sensors: they are cheap, provide color and geometric information for scene understanding, and consume less power. Many computer-vision algorithms are available for position estimation. For example, visual odometry (VO) estimates the ego-motion of a vehicle with an onboard camera. VO incrementally estimates the vehicle's pose by examining the changes that motion induces on the input images [6]. Using a red, green, blue, depth (RGBD) camera, the method reported in [7] improved the VO algorithm by including a novel covariance estimation technique. The resulting VO-based algorithm allowed autonomous quadrotor navigation with satisfactory results. The depth camera measurement allows for determining the position scale factor.

The SLAM algorithm estimates the vehicle's pose and, at the same time, constructs a map of the surroundings. The most successful versions of SLAM, running in real time, are ORB-SLAM [8] and LSD-SLAM [9]. These SLAM variants rely on techniques used to calculate the camera position and construct the map, such as feature extraction, or direct methods that operate on the image intensities [10]. A comparison between SLAM algorithms for mobile robot navigation in indoor environments is reported in [11], where it is concluded that ORB-SLAM can be used to determine the robot position with an additional module to recover the scale factor. The report [12] presents quadrotor autonomous navigation using a SLAM algorithm without determining the position scale factor.

Semidirect visual odometry (SVO) is a hybrid algorithm combining feature-based and direct methods. It estimates the relative motion between two frames by minimizing photometric errors. The projection error between the location of the feature points and their predicted positions is minimized to obtain the optimal camera pose [13]. The autonomous navigation of a UAV using SVO and a recovery mechanism to reinitialize the visual map when a failure occurs are proposed in [14]. The navigation strategy includes a pose estimation scheme for temporary vehicle control and a method to correct the scene scale factor using altitude measurements.

With some differences, computer-vision algorithms can be implemented using monocular or stereoscopic cameras. Stereo cameras can capture three-dimensional images, meaning the scene's depth is known. This ability leads to better accuracy and resolution than in the monocular camera case for position estimation purposes. The algorithms ORB-SLAM, LSD-SLAM, and SVO, among others, were first developed for monocular camera implementation and, years after, improved for multi-camera configurations as reported in [15–17], respectively.

Nevertheless, monocular cameras are preferred for implementation in small vehicles for several reasons: a single camera is easy to mount due to its smaller size, is lighter and cheaper, and consumes less power. Additionally, a single camera configuration is free from the burden of multi-camera calibration and requires less processing power from the CPU onboard than multi-camera configurations [18]. Only one drawback is present for monocular cameras: they cannot recover the image's three-dimensional structure and the camera position with complete metric information; in other words, the information on the scene's depth is unavailable. This phenomenon is known as similarity ambiguity [19]. At least one piece of metric information is required to recover the absolute scale factor. This cue may come from prior scene knowledge, such as camera height, object size, vehicle speed, stereo camera baseline, or other sensors such as LiDAR or GPS.

Some methods have been proposed to deal with the similarity ambiguity problem. In [20,21], the extra piece of geometric information to determine the position measurement scale factor comes from an ultrasonic sensor and a one-dimensional laser range finder (LRF), respectively. These approaches require additional sensors onboard. Besides, the absolute scale is only calculated on the axis where the sensor is mounted, so it is assumed that the scale is the same on the other two axes, which is not always valid.

An EKF algorithm considering multirotor dynamics is proposed in [22] to estimate the scale factor online. A scale factor observability analysis supports the estimator design.

However, using EKF on this approach makes the estimator nondeterministic, so stability is not formally proven.

In the field of deterministic estimators, ref. [23] presents a scale estimator based on control stability. It shows that the absolute scale and control gain have a unique linear relationship. The absolute scale can be estimated by detecting self-induced oscillations and analyzing the system stability. The problem with this approach is that an adaptive control technique must be used for an online estimation, leaving out other types of controllers.

This article presents a scale factor estimator in the cartesian plane fused with a velocity observer, deterministic and based on the quadrotor dynamic model, using only onboard sensors. The set of sensors provides the quadrotor's attitude and angular velocity from an attitude and heading reference system (AHRS), the quadrotor's acceleration from the set of sensors of the inertial measurement unit, and the scaled position from a SLAM algorithm based on a monocular camera. The scale factor estimator and the velocity observer are designed following the immersion and invariance methodology introduced in [24]. The singular contributions of this work are: the estimator and observer are designed considering the full quadrotor nonlinear model, the Coriolis forces are not neglected, the position scale factor is reconstructed in all three dimensions, and it is formally proven using Lyapunov arguments that the estimator and observer errors locally asymptotically converge to zero. Numerical simulations using Gazebo are presented to support the theoretical developments. The outcomes of this paper are based on the preliminary works reported in [25,26].

The remaining parts of the paper are arranged as follows. The sensor models used are presented in Section 2, along with the quadrotor dynamics. Section 3 outlines the fundamental contribution of this paper and describes the mathematical advancements used to create the scale factor estimator. Through numerical simulations, Section 4 illustrates the performance of the estimator. Finally, Section 5 wraps up this paper with a few closing thoughts and suggestions for future work.

2. Materials and Methods

Table 1 summarizes the notation used to introduce the quadrotor dynamic model.

Table 1. Quadrotor dynamic model notation.

Symbol	Variable	Units
$X = [x \ y \ z]^\top$	Translational position in inertial coordinates	m
$R \in SO(3)$	Rotation matrix from body to inertial coordinates	dimensionless
$V^b = [u \ v \ w]^\top$	Translational velocity in body frame coordinates	m/s
$\Omega = [p \ q \ r]^\top$	Angular velocity in body coordinates	rad/s
$M^b = [M_x^b \ M_y^b \ M_z^b]^\top$	Moments generated by the differential thrust, and reaction moment between the four rotors	Nm
m	Quadrotor mass	kg
g	Gravity acceleration constant	m/s ²
T_T	Total thrust generated by the four rotors	N
μ	Parameter related to aerodynamic drag force [27]	kg/s
$J \in \mathbb{R}^{3 \times 3}$	Quadrotor inertia matrix	kg m ²

In Table 1, the International System of Units is considered, and

$$SO(3) = \left\{ R \in \mathbb{R}^{3 \times 3} \mid R^\top R = I, \det(R) = 1 \right\}$$

with I the identity matrix.

2.1. Quadrotor Dynamics

An inertial coordinate frame and a non-inertial coordinate frame (body frame) attached to the quadrotor center of gravity are needed to describe the quadrotor dynamic, see

Figure 1. The following equations, expressed in mixed inertial and body coordinates, describe the translational and rotational quadrotor dynamics [26]:

$$\begin{aligned}\dot{X} &= RV^b \\ m\dot{V}^b &= mgR^\top e_3 - T_T e_3 - \mu HV^b - mS(\Omega)V^b \\ \dot{R} &= RS(\Omega) \\ J\dot{\Omega} &= -S(\Omega)J\Omega + M^b\end{aligned}\quad (1)$$

with

$$e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, S(\Omega) = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}$$

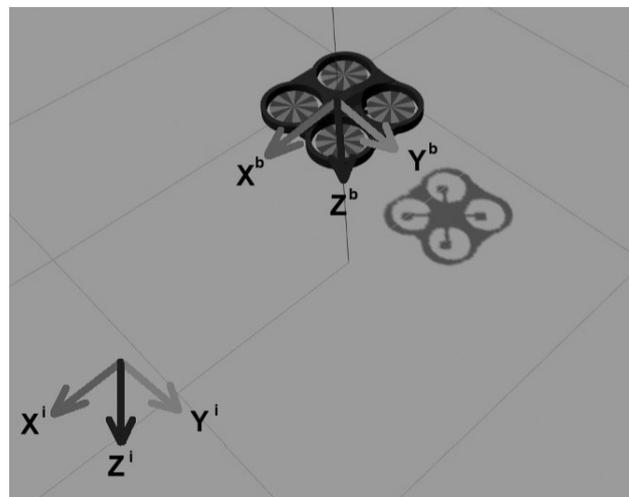


Figure 1. Inertial $X^i Y^i Z^i$ and Body $X^b Y^b Z^b$ coordinates.

2.2. Available Sensors

It is assumed that the quadrotor carries onboard a set of sensors that provide the following measurements.

2.2.1. Scaled Position

The vehicle carries a monocular camera facing the horizontal plane and the necessary computer power to implement a monocular-vision algorithm to determine its scaled inertial position. Therefore, the following measurement is available

$$y_1 = X_s = \begin{bmatrix} k_x x \\ k_y y \\ k_z z \end{bmatrix} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \text{diag}(K)X \quad (2)$$

where X_s is the scaled position delivered by the monocular SLAM vision algorithm and $K = [k_x \ k_y \ k_z]^\top$ is the dimensionless unknown scale factor on the axes $X^i Y^i Z^i$, respectively.

Remark 1. The operator $\text{diag}(A)$ represents a diagonal matrix whose elements are the elements of vector $A \in \mathbb{R}^3$. This operator satisfies the following identities

$$\text{diag}(A + B) = \text{diag}(A) + \text{diag}(B) \quad (3)$$

$$\text{diag}(A)B = \text{diag}(B)A \quad (4)$$

with $A, B \in \mathbb{R}^3$ vectors.

2.2.2. Specific Acceleration

Commonly, quadrotors are equipped with an inertial measuring unit (IMU) that measures the Earth's magnetic field intensity, angular velocity, and specific acceleration in body coordinates. According to the Accelerometer Tutorial reported in [27], the specific acceleration measured by an accelerometer mounted on a quadrotor is given by

$$a^b = \frac{1}{m} F_T^b - gR^\top e_3 \quad (5)$$

where a^b is the specific acceleration measured in the body axis and F_T^b is the total external force acting on the quadrotor expressed in the body axis. From the quadrotor dynamics model in Equation (1), it follows that

$$F_T^b = mgR^\top e_3 - T_T e_3 - \mu HV^b \quad (6)$$

as a result,

$$a^b = -\frac{T_T}{m} e_3 - \frac{\mu}{m} HV^b \quad (7)$$

Hence, the specific acceleration is an available output, element wise it reads as

$$y_2 = \begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} = -\frac{1}{m} \begin{bmatrix} \mu u \\ \mu v \\ T_T \end{bmatrix} \quad (8)$$

with a_x^b , a_y^b and a_z^b the specific acceleration along the body axis.

2.2.3. Attitude and Heading Reference Systems

The device that computes the quadrotor's attitude and rotational velocity from the IMU measurements is called the attitude and heading reference system (AHRS). Assuming that the quadrotor carries an AHRS, the following signals are available.

$$y_3 = R = \begin{bmatrix} r_1^\top \\ r_2^\top \\ r_3^\top \end{bmatrix} \quad (9)$$

$$y_4 = \Omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (10)$$

where $r_i \in \mathbb{S}^2$, $i = 1, \dots, 3$ are the columns of the rotation matrix transposed,

$$\mathbb{S}^2 = \{A \in \mathbb{R}^3 \mid A^\top A = 1\}$$

is the unit 2-sphere.

2.2.4. Vertical Speed

Through the use of a laser sensor or an ultrasonic sensor, the vertical quadrotor position can be measured so that the vertical speed can be determined. As a result, it is presumed that the subsequent measurement is available

$$y_5 = w \quad (11)$$

Finally, note that the quadrotor translational dynamic, the first equation in (1), expressed in terms of the measured states reads as

$$\dot{V}^b = gy_3^\top e_3 + y_2 - S(y_4)V^b \quad (12)$$

2.3. Immersion and Invariance Observers

The following developments are based on Chapter 5 of [24]. Consider the following non-linear, deterministic, time invariant system

$$\begin{aligned}\dot{\eta} &= f_1(\eta, y) \\ \dot{y} &= f_2(\eta, y)\end{aligned}\quad (13)$$

where $\eta \in \mathcal{R} \subset \mathbb{R}^n$ and $y \in \mathcal{Y} \subset \mathbb{R}^m$ are the unmeasured and measured states, respectively. It is assumed that the vector fields $f_1(\eta, y)$ and $f_2(\eta, y)$ are forward complete.

Definition 1. *The dynamic system*

$$\dot{\hat{\eta}} = \varphi(\hat{\eta}, y) \quad (14)$$

with $\hat{\eta} \in \mathbb{R}^n$, is an observer for the unmeasured state η if there exists a mapping $\beta : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ such that the manifold

$$\mathcal{M} = \{(\eta, \hat{\eta}, y) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \mid \beta(\hat{\eta}, y) = \eta\}$$

has the following properties

- \mathcal{M} is positively invariant,
- All trajectories of (13), (14) that start in a neighbourhood of \mathcal{M} asymptotically converge to \mathcal{M} .

The construction of the observer of the form given in Definition 1 requires additional properties on the mapping $\beta(\hat{\eta}, y)$, as stated in the following result.

Theorem 1. *Consider the system (13). Suppose that there exist differentiable maps $\beta : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ such that*

A1 For all $\hat{\eta}$ and y the map $\beta(\hat{\eta}, y)$ satisfies

$$\det\left(\frac{\partial \beta}{\partial \hat{\eta}}\right) \neq 0$$

A2 The dynamic system

$$\dot{\tilde{\eta}} = f_1(\tilde{\eta} + \beta(\hat{\eta}, y), y) - f_1(\beta(\hat{\eta}, y), y) - \frac{\partial \beta}{\partial y}(f_2(\tilde{\eta} + \beta(\hat{\eta}, y), y) - f_2(\beta(\hat{\eta}, y), y)) \quad (15)$$

has a (globally) asymptotically stable equilibrium at $\tilde{\eta} = 0$ uniformly in $\hat{\eta}$ and y .

Then, the system (14) with

$$\varphi(\hat{\eta}, y) = \left(\frac{\partial \beta}{\partial \hat{\eta}}\right)^{-1} \left(f_1(\beta(\hat{\eta}, y), y) - \frac{\partial \beta}{\partial y} f_2(\beta(\hat{\eta}, y), y)\right) \quad (16)$$

is a (global) observer for (13).

The proof of this Theorem is presented in Appendix A. The result expressed in Theorem 1 is followed to design the velocity observer and the scale factor estimator.

3. Observer and Estimator Design

This section discusses how the observer and the estimator that reconstruct V^b and K , respectively, are designed from the available measurements of acceleration, scaled position, attitude, angular velocity, and vertical speed.

3.1. Observation and Estimation Problems

The following terms state the observation problem. Assume that the outputs y_i , $i = 1, \dots, 5$ are measurable. Design two dynamic systems, likely, of the form

$$\begin{aligned}\dot{\hat{V}}^b &= \varphi_1(\hat{V}^b, y_2, y_3, y_4, y_5) \\ \dot{\hat{K}} &= \varphi_2(\hat{K}, \hat{V}^b, y_1, y_2, y_3, y_4, y_5)\end{aligned}$$

where $\hat{V}^b \in \mathbb{R}^3$ and $\hat{K} \in \mathbb{R}^3$, such that two functions exist, $\beta_1 \in \mathbb{R}^3$ and $\beta_2 \in \mathbb{R}^3$, that depend on the available information, and the following identities asymptotically hold

$$\begin{aligned}\lim_{t \rightarrow \infty} \beta_1(\hat{V}^b, y_2, y_3, y_4, y_5) &= V^b \\ \lim_{t \rightarrow \infty} \beta_2(\hat{K}, \hat{V}^b, y_1, y_2, y_3, y_4, y_5) &= K\end{aligned}\quad (17)$$

3.2. Velocity Observer

According to the immersion and invariance technique, the observation error is defined as follows

$$\tilde{V}^b = V^b - \beta_1(\hat{V}^b, \sigma) \quad (18)$$

with β_1 element wise reading as

$$\beta_1 = \begin{bmatrix} \beta_{1x} \\ \beta_{1y} \\ \beta_{1z} \end{bmatrix} \quad (19)$$

where

$$\dot{\sigma} = \begin{bmatrix} a_x^b \\ a_y^b \\ -w \end{bmatrix} = \begin{bmatrix} -\frac{\mu}{m}u \\ -\frac{\mu}{m}v \\ -w \end{bmatrix} = -\tilde{H}V^b \quad (20)$$

and

$$\tilde{H} = \begin{bmatrix} \frac{\mu}{m} & 0 & 0 \\ 0 & \frac{\mu}{m} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Equation (18) models the distance to the manifold \mathcal{M} of Definition 1, where the velocity V^b is equal to $\beta_1(\hat{V}^b, \sigma)$. This distance must asymptotically converge to zero to complete the observer design. Note that the output y_2 is not directly used since the time derivative of \tilde{V}^b will require the computation of \dot{y}_2 ; this is the reason why the new state σ is introduced.

The time derivative of \tilde{V}^b is given by

$$\dot{\tilde{V}}^b = \dot{V}^b - \frac{\partial \beta_1}{\partial \hat{V}^b} \dot{\hat{V}}^b - \frac{\partial \beta_1}{\partial \sigma} \dot{\sigma} \quad (21)$$

Substituting \dot{V}^b , the body velocity V^b and the time derivative of σ from Equations (12), (18), and (20), respectively, one has

$$\dot{\tilde{V}}^b = gy_3^\top e_3 + y_2 - S(y_4)(\tilde{V}^b + \beta_1) - \frac{\partial \beta_1}{\partial \hat{V}^b} \dot{\hat{V}}^b + \frac{\partial \beta_1}{\partial \sigma} \tilde{H}(\tilde{V}^b + \beta_1) \quad (22)$$

Now, the observer state dynamic $\dot{\hat{V}}^b$ is defined in terms of the known signals, as

$$\dot{\hat{V}}^b = \frac{\partial \beta_1}{\partial \hat{V}^b}^{-1} \left(gy_3^\top e_3 + y_2 - S(y_4)\beta_1 + \frac{\partial \beta_1}{\partial \sigma} \tilde{H}\beta_1 \right) \quad (23)$$

Substituting (23) into (22), one obtains

$$\dot{\tilde{V}}^b = -S(y_4)\tilde{V}^b + \frac{\partial\beta_1}{\partial\sigma}\tilde{H}\tilde{V}^b \quad (24)$$

Consider the following Lyapunov function to define the function $\beta_1(\hat{V}^b, \sigma)$

$$v_V = \frac{1}{2}\tilde{V}^{b\top}\tilde{V}^b \quad (25)$$

it follows that

$$\dot{v}_V = \tilde{V}^{b\top}\frac{\partial\beta_1}{\partial\sigma}\tilde{H}\tilde{V}^b \quad (26)$$

Hence, to guarantee that the time derivative of v_V is negative-definite, the matrix $\frac{\partial\beta_1}{\partial\sigma}\tilde{H}$ must also be negative-definite. On the other hand, Equation (23) requires the matrix $\frac{\partial\beta_1}{\partial\hat{V}^b}$ to be invertible. Note that selecting

$$\beta_1(\hat{V}^b, \sigma) = \hat{V}^b - \Gamma_v\sigma \quad (27)$$

with

$$\Gamma_v = \begin{bmatrix} \gamma_{vx} & 0 & 0 \\ 0 & \gamma_{vy} & 0 \\ 0 & 0 & \gamma_{vz} \end{bmatrix} \quad (28)$$

and γ_{vx} , γ_{vy} and γ_{vz} positive gains, it follows that

$$\dot{v}_V = -\tilde{V}^{b\top}\Gamma_v\tilde{H}\tilde{V}^b = -\tilde{V}^{b\top}diag(h)\tilde{V}^b, \quad \frac{\partial\beta_1}{\partial\hat{V}^b} = I_3 \quad (29)$$

with $h = \left[\frac{\gamma_{vx}\mu}{m} \quad \frac{\gamma_{vy}\mu}{m} \quad \gamma_{vz} \right]^\top$ and I_3 the 3×3 identity matrix. As a result, $\frac{\partial\beta_1}{\partial\sigma}\tilde{H}$ is negative-definite and $\frac{\partial\beta_1}{\partial\hat{V}^b}$ is invertible.

3.3. Scale Factor Estimator

In reference [25], the scale factor estimator needs the translational velocity as a measurable output, but in this work it is available through the observer designed (27) according to (17). It is important to note also in this equation that β_2 depends on states expressed in mixed inertial and body coordinates, unlike β_1 which depends only on states expressed in body coordinates. In order to have all the states expressed in inertial coordinates, β_1 needs to be translated with the rotation matrix. The inertial velocity is introduced as follows

$$V^i = y_3V^b \quad (30)$$

Additionally, the inertial velocity observer error is defined

$$\tilde{V}^i = V^i - \hat{V}^i \quad (31)$$

with

$$\hat{V}^i = y_3\beta_1 \quad (32)$$

Now, the scale factor estimation error is defined in the following form

$$\tilde{K} = K - \beta_2(\hat{K}, y_1, \hat{V}^i) \quad (33)$$

with β_2 element wise reading as

$$\beta_2 = \begin{bmatrix} \beta_{2x} \\ \beta_{2y} \\ \beta_{2z} \end{bmatrix} \quad (34)$$

The derivative with respect to the time of the estimation error is

$$\dot{\hat{K}} = -\frac{\partial\beta_2}{\partial\hat{K}}\dot{K} - \frac{\partial\beta_2}{\partial y_1}\dot{y}_1 - \frac{\partial\beta_2}{\partial\hat{V}^i}\dot{\hat{V}}^i \quad (35)$$

From Equations (1) and (2), it follows that

$$\begin{aligned} \dot{y}_1 &= \text{diag}(K)y_3V^b = \text{diag}(K)V^i \\ \dot{y}_3 &= y_3S(y_4) \end{aligned} \quad (36)$$

By combining Equations (18) and (32), one obtains

$$\hat{V}^i = y_3(V^b - \tilde{V}^b) \quad (37)$$

thus,

$$\dot{\hat{V}}^i = ge_3 + y_3y_2 - \Gamma_v\tilde{H}\tilde{V}^i \quad (38)$$

Substituting (36) and (37) into (35), one obtains

$$\dot{\hat{K}} = -\frac{\partial\beta_2}{\partial\hat{K}}\dot{K} - \frac{\partial\beta_2}{\partial y_1}\text{diag}(K)V^i - \frac{\partial\beta_2}{\partial\hat{V}^i}(ge_3 + y_3y_2 - \Gamma_v\tilde{H}\tilde{V}^i) \quad (39)$$

Now, substituting K from (33) and V^i from (31), the scale factor estimation error becomes

$$\dot{\hat{K}} = -\frac{\partial\beta_2}{\partial\hat{K}}\dot{\hat{K}} - \frac{\partial\beta_2}{\partial y_1}\text{diag}(\tilde{K} + \beta_2)(\tilde{V}^i + \hat{V}^i) - \frac{\partial\beta_2}{\partial\hat{V}^i}(ge_3 + y_3y_2 - \Gamma_v\tilde{H}\tilde{V}^i) \quad (40)$$

The dynamic of the scale factor estimator state is defined in terms of the known signals as follows

$$\dot{\hat{K}} = \frac{\partial\beta_2}{\partial\hat{K}}^{-1} \left(-\frac{\partial\beta_2}{\partial y_1}\text{diag}(\beta_2)\hat{V}^i - \frac{\partial\beta_2}{\partial\hat{V}^i}(ge_3 + y_3y_2) \right) \quad (41)$$

After substituting $\dot{\hat{K}}$ into the scale factor estimator error (40), it follows that

$$\dot{\hat{K}} = -\frac{\partial\beta_2}{\partial y_1}(\text{diag}(\tilde{K})\hat{V}^i + \text{diag}(\tilde{K} + \beta_2)\tilde{V}^i) + \frac{\partial\beta_2}{\partial\hat{V}^i}\Gamma_v\tilde{H}\tilde{V}^i \quad (42)$$

Once again, the function β_2 needs to be defined to ensure that the estimation error \tilde{K} converges to zero with $\frac{\partial\beta_2}{\partial\tilde{K}}$ an invertible matrix. Thus, the following vector function is proposed

$$\beta_2(\hat{K}, y_1, \hat{V}^i) = \hat{K} + \Gamma_k\text{diag}(y_1)\hat{V}^i = \hat{K} + \Gamma_k\text{diag}(\hat{V}^i)y_1 \quad (43)$$

with

$$\Gamma_k = \begin{bmatrix} \gamma_{kx} & 0 & 0 \\ 0 & \gamma_{ky} & 0 \\ 0 & 0 & \gamma_{kz} \end{bmatrix} \quad (44)$$

and $\gamma_{ki}, i = x, y, z$ the scale factor estimator gains.

Replacing (43) into (42), one obtains

$$\dot{\hat{K}} = -\Gamma_k\text{diag}(\hat{V}^i)(\text{diag}(\hat{V}^i)\tilde{K} + \text{diag}(\tilde{V}^i)K) + \Gamma_k\text{diag}(y_1)\Gamma_v\tilde{H}\tilde{V}^i \quad (45)$$

where (3), (4) and (33) had been considered. From (31), it follows that

$$\begin{aligned}\dot{\tilde{K}} &= -\Gamma_k \left(\text{diag}(V^i) \text{diag}(V^i) + \text{diag}(\tilde{V}^i) \text{diag}(\tilde{V}^i) \right) \tilde{K} \\ &\quad + 2\Gamma_k \text{diag}(V^i) \text{diag}(\tilde{V}^i) \tilde{K} - \Gamma_k \left(\text{diag}(V^i) - \text{diag}(\tilde{V}^i) \right) \text{diag}(\tilde{V}^i) K \\ &\quad + \Gamma_k \text{diag}(y_1) \Gamma_v \tilde{H} \tilde{V}^i\end{aligned}\quad (46)$$

The following assumptions are considered to state the main result of this paper.

Assumption 1. *The following identity holds.*

$$\lim_{t \rightarrow \infty} \int_0^t \text{diag}(V^i(\tau)) \text{diag}(V^i(\tau)) d\tau = \infty I$$

with $I \in \mathbb{R}^{3 \times 3}$ the identity matrix.

Assumption 2. *There exist control inputs T_T and M^b such that the following quadrotor states can be upper bounded, this is*

$$\|V^i\| \leq \kappa_0, \quad \|y_1\| \leq \kappa_1 \quad (47)$$

for some not-necessarily constants κ_0 and κ_1 . The notation $\|(\cdot)\|$ stands for the Euclidean norm for a matrix or vector (\cdot) .

Remark 2. *Assumption 1 is the persistence of the excitation condition; in this case, fulfilling this condition implies that the quadrotor must move to estimate the scale factor successfully. Assumption 2 means that a control loop allows the quadrotor to fly stably; consequently, the quadrotor dynamics is forward complete.*

The following Proposition summarizes the main result of this work.

Proposition 1. *Under Assumptions 1 and 2, there are matrix gains Γ_v and Γ_k such that the dynamic systems (23) and (41) are local observers for the translational velocity and estimators of the scale factor. The translational velocity and scaling factor are rebuilt in (27) and (43).*

The proof of this Proposition is reported in Appendix B.

4. Numerical Simulations

A numerical simulation study was performed on different platforms to evaluate the observer and estimator's performance.

4.1. Matlab-Simulink

The first one was performed using Matlab-Simulink, to avoid problems such as sensors' noise and external disturbances so that we can evaluate the estimators by themselves. A program was designed to simulate a quadrotor in a closed loop with the controller developed in [28], tracking a circular trajectory on the Cartesian plane and a sinusoidal form on the vertical plane. To fulfill Assumption 1, the desired trajectories were $x_d = A \cos(\omega t)$, $y_d = A \sin(\omega t)$ and $z_d = A \cos(\omega t)$. For the velocity observer, the initial conditions used were $V^b(0) = [-0.2 \ 0.3 \ 0.2]^\top$ with a proposed $\mu = 0.6$ and gains $\gamma_{vx} = 1.2$, $\gamma_{vy} = 1.2$ and $\gamma_{vz} = 1.2$. Regarding the monocular-vision positioning algorithm for this numerical simulation, any real values for the scale factor K can be used; nevertheless, in more realistic simulations such as the ones in the next section, it is observed that the scaled position is always smaller than the real position, $X_s < X$, so $K < 1$; hence, the real values of the scale factor for this simulation were fixed at $K = [0.65 \ 0.7 \ 0.55]^\top$. The gains used for the estimator were $\gamma_{kx} = 2.0$, $\gamma_{ky} = 2.0$ and $\gamma_{kz} = 2.0$.

Figure 2 shows the velocity observer error \tilde{V}^b for this simulation, where it can be seen that the velocities converge to zero correctly. Note that the velocity error on the Z axis, \tilde{w} , is always zero because the speed on this axis is measurable, (11).

Figure 3 shows the scale factor estimator error \tilde{K} that also proves the correct convergence of the estimator. In this graph the cascade behavior of (46) with (24) can also be seen, which means that \tilde{K} will always converge after \tilde{V}^b due to the interconnection term Ψ .

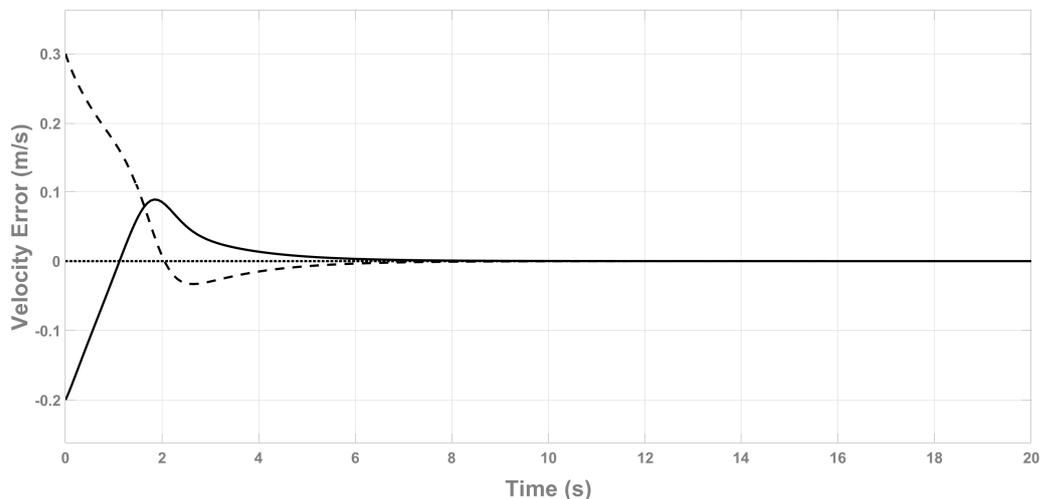


Figure 2. Velocity observer error \tilde{V}^b . \tilde{u} (continuous line), \tilde{v} (dashed line), \tilde{w} (dotted line).

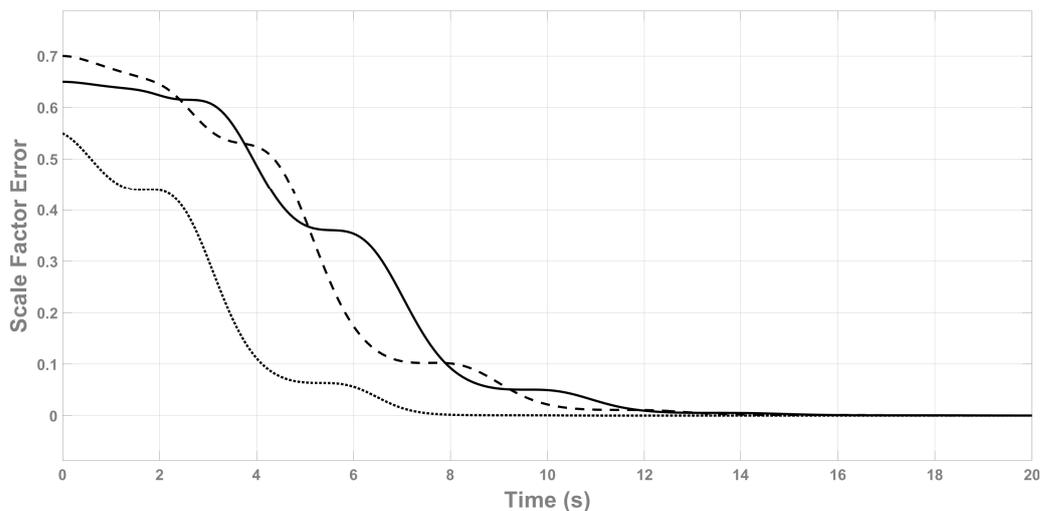


Figure 3. Scale factor estimator error \tilde{K} . \tilde{k}_x (continuous line), \tilde{k}_y (dashed line), \tilde{k}_z (dotted line).

4.2. Gazebo

It is time to put the observers to the test in a simulator, such as Gazebo, which is more like reality after confirming their proper operation in a controlled setting. The Gazebo is an open-source 3D robotics simulator. It incorporates the open dynamics engine (ODE) as a physics engine, OpenGL for graphics, support code for sensor simulation, and actuator control. The robot operating system (ROS) and the Gazebo simulator are wholly linked.

In the previous simulation, it was easy to set simulation values for the constant μ and the scaled position X_s delivered by a monocular-vision algorithm. In the Gazebo simulation, such values will have to be treated more rigorously as would be performed in an actual experiment.

4.2.1. Calculation of μ

μ is a term related to the drag coefficient, a positive constant representing a combination of the profile and induced drag forces on the rotors, known in the helicopter literature as “rotor drag” [27,29]. Like any other parameter of the quadrotor, such as its weight (m), it must be measured before experiments. Due to the units of this parameter, listed in Table 1, it is known as the “mass flow rate”.

From Equation (8), it can be seen that

$$\begin{aligned} a_x^b &= -\frac{\mu}{m}u \\ a_y^b &= -\frac{\mu}{m}v \end{aligned} \quad (48)$$

From any of these two equations, the constant μ can be measured by having access to the accelerometer and translational velocities measurements, for example

$$\mu = -\frac{a_x^b}{u}m \quad (49)$$

We have access to both of these measurements in Gazebo, so a circular trajectory tracking simulation was performed to measure these states. The calculated value for the quadrotor used in Gazebo is $\mu = 0.18$.

With μ calculated, Figure 4 shows the Equation (48) on the X axis with the quadrotor following a circular trajectory, where it can be seen that the relation holds, so the calculated μ is correct. Note that the accelerometer readings (a_x^b) have noise added due to the rotors.

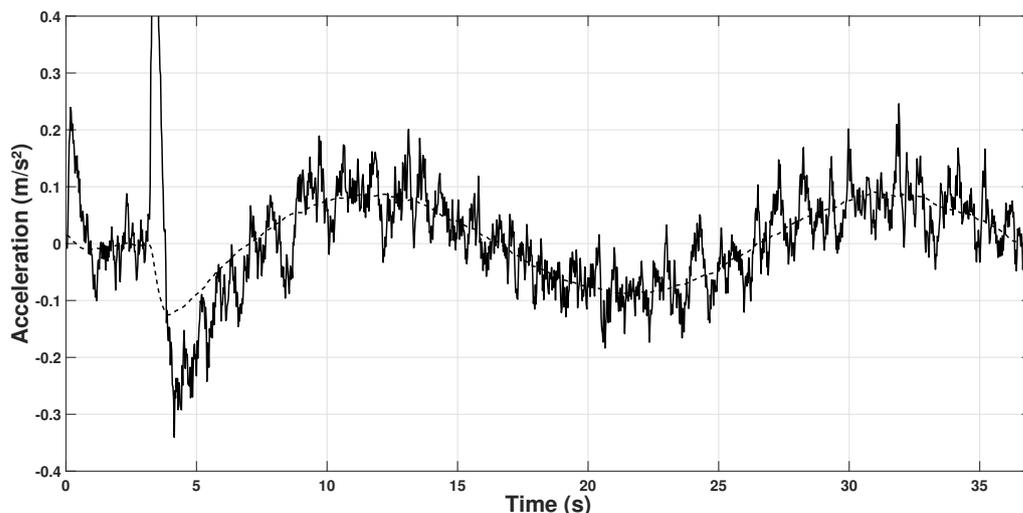


Figure 4. Relation between accelerometer and μ on the X^b axis. a_x^b (continuous line), $-\frac{\mu}{m}u$ (dashed line).

4.2.2. Monocular-Vision Algorithm

As mentioned before, the designed scale factor estimator works with any computer-vision algorithm that implements a monocular camera and delivers position measurements. For this simulation, we use ORB-SLAM2 [15] due to its precision and accuracy, it is easy to install and well-documented, and it has ROS integration. Hence, it is easy to add to the Gazebo environment.

4.2.3. Trajectory Tracking Control Using the Scale Factor Estimator

The simulation involves flying the quadrotor running ORB-SLAM2 from the visual information obtained by its onboard monocular camera without position control following a diagonal trajectory in the horizontal plane to obtain the information required to make the

identities in (17) hold. After this step, the scale factor K will be known through β_2 , so the quadrotor will be able to measure its actual position with y_1 , (2). Then, the quadrotor will follow a lemniscate trajectory autonomously (closed-loop).

The low-level controller of the quadrotor is driven by [30], which is a driver to interface with Parrot AR-Drone quadrotors through ROS. It takes translational velocities as control inputs.

Figure 5 shows the Gazebo environment with the SLAM algorithm running. The window on the left shows what the onboard camera is seeing with the features (points) detected, and the window on the right depicts the mapping construction.

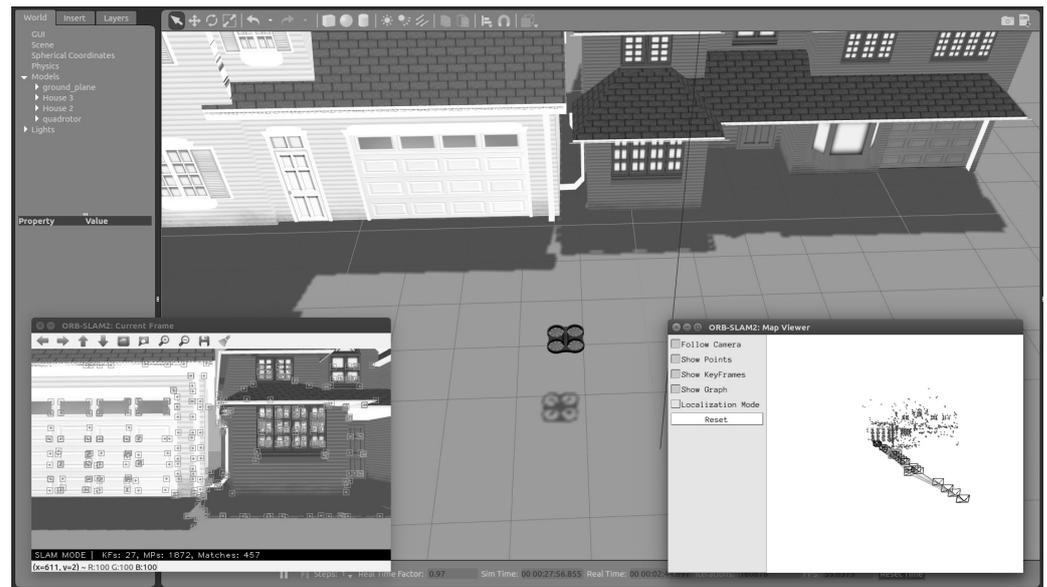


Figure 5. Gazebo environment.

The first step is to fly the quadrotor in open-loop over a diagonal trajectory from position $(0,0)$ to $(3,3)$, after that to the position $(-3,-3)$ and finally back to the origin $(0,0)$, always maintaining a constant velocity of $u_d = 0.2$ and $v_d = 0.2$.

Figures 6 and 7 show the real velocity and the estimated by the observer. With the initial conditions of the real velocity and the observer being the same, $V^b(0) = [0 \ 0 \ 0]^T$, and the high gains $\gamma_{vx} = 30$, $\gamma_{vy} = 30$ and $\gamma_{vz} = 30$, make the observer converge immediately.

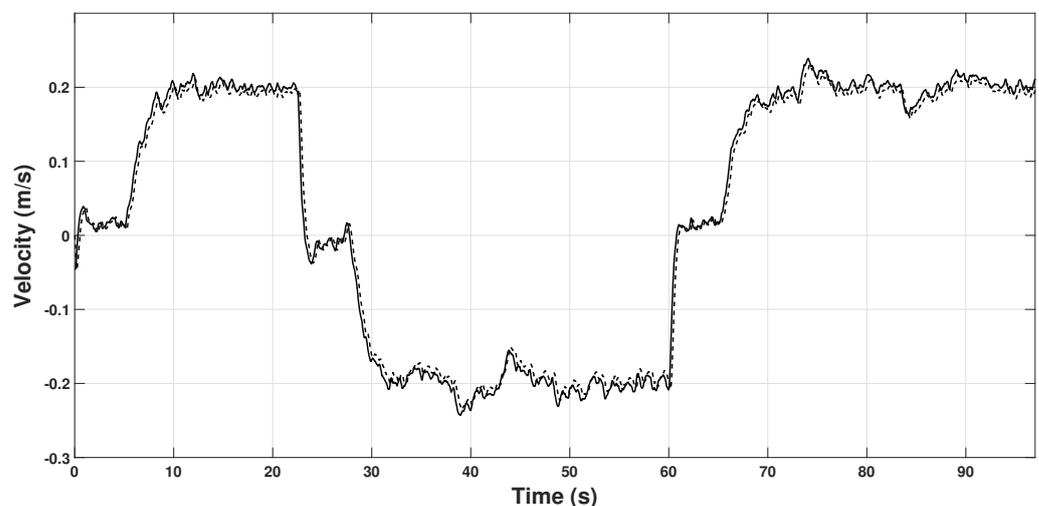


Figure 6. Observed speed on the X^b axis. u (continuous line), β_{1x} (dashed line).

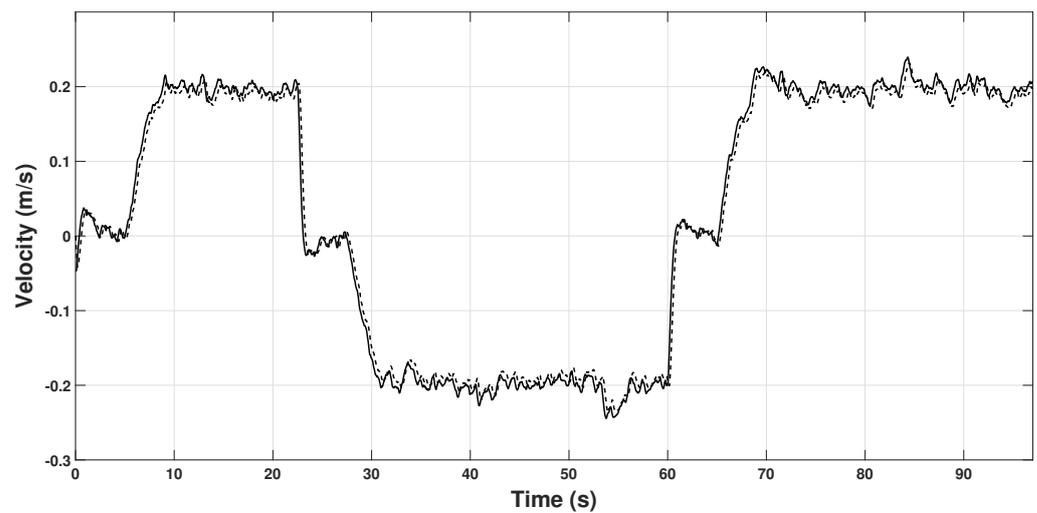


Figure 7. Observed speed on the Y^b axis. v (continuous line), β_{1y} (dashed line).

Figures 8 and 9 show the scale factor estimation on the X and Y axes, respectively. Note that from time 0 s to 6 s, the estimator remains in zero because, in that period, the quadrotor takes off and holds in hover for some seconds, so Assumption 1 is not fulfilled. In the period 22 s to 60 s, some peaks appear because the quadrotor reaches the coordinates $(3, 3)$ and $(-3, -3)$, respectively. Hence, it changes velocity abruptly to change its direction of movement to reach the next point. These changes in velocity can also be seen in Figures 6 and 7 in the time periods.

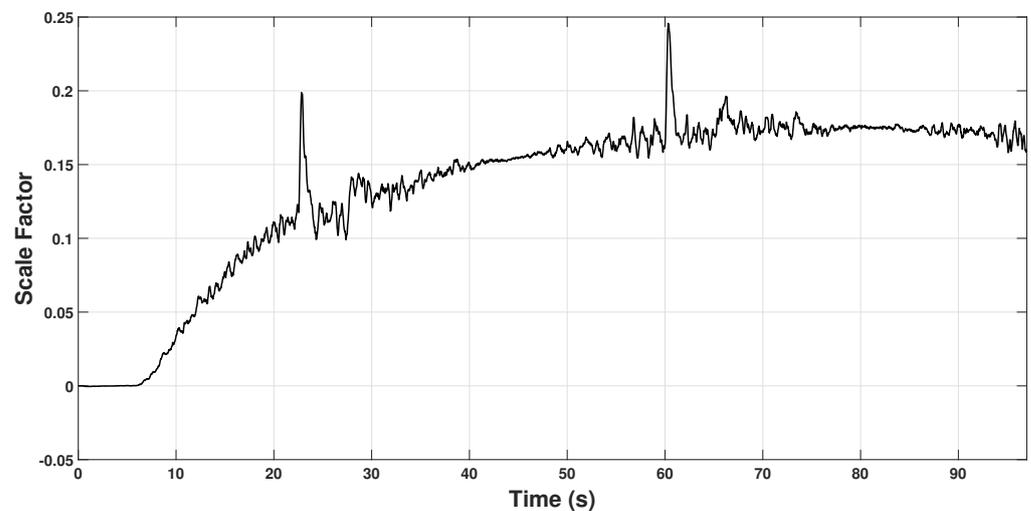


Figure 8. Estimated scale factor β_{2x} on the X^i axis.

After the open-loop routine, β_2 is calculated to converge at $k_x = 0.175$ and $k_y = 0.13$. It can also be seen in Figures 8 and 9 that \tilde{K} converges exponentially, as it was anticipated in (A7). From this point, knowing the scale factor K , the quadrotor will be able to measure its actual position through y_1 in (2) as follows

$$X = \text{diag}(K)^{-1}y_1 \quad (50)$$

Finally, using the estimated information, the quadrotor will follow a lemniscate trajectory autonomously in closed-loop; that is, $x_d = 1.5\cos(\omega t)$ and $y_d = \sin(2\omega t)$. Figures 10 and 11 show the real position and the position measured by the quadrotor on the X and Y axis, respectively, along the lemniscate trajectory.

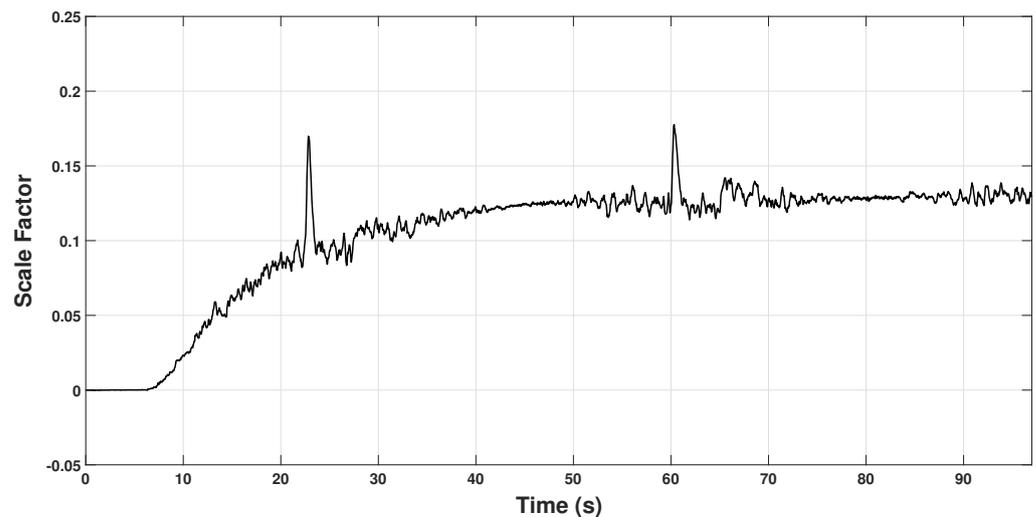


Figure 9. Estimated scale factor β_{2y} on the Y^i axis.

Figure 12 shows the XY graph of the trajectory validating the closed-loop control of the quadrotor using the proposed scale factor estimator. The quadrotor completed the whole lemniscate trajectory twice with the given simulation time. The big arrow indicates the direction the onboard camera faces the whole trajectory, the line on the left represents the buildings' location, and the small arrows indicate the quadrotor motion direction.

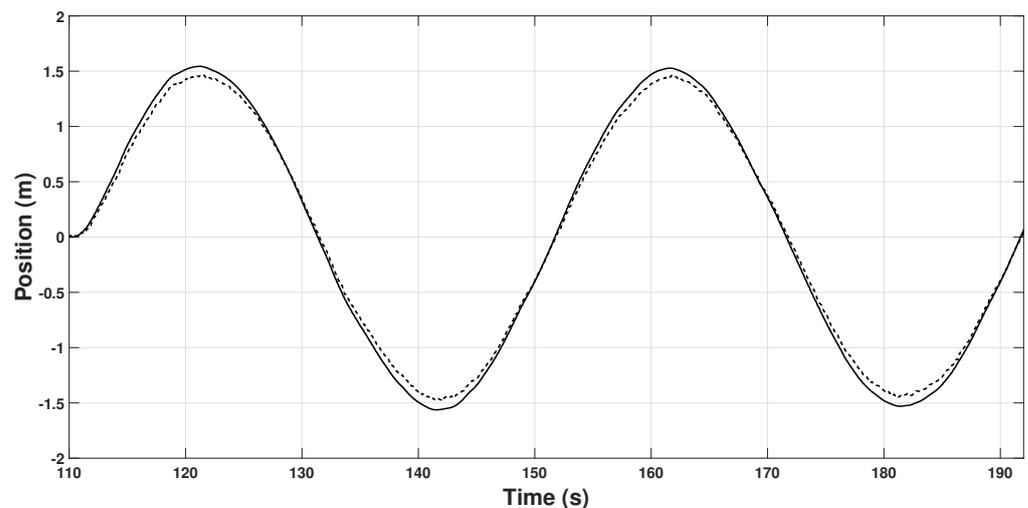


Figure 10. Quadrotor position on the X^i axis. x (continuous line), $\frac{x_s}{k_x}$ (dashed line).

As shown in Figure 12, the SLAM algorithm has better precision in the left part of the lemniscate because the camera is closer to the buildings, so more image features fed the algorithm. On the diagonals, the SLAM algorithm performs better when the quadrotor moves towards the buildings than when it moves away. Although the scale factor estimator recovers the actual position dimension, it does not help in any sense to improve the SLAM algorithm performance.

Changing the position of the house on the right in the Gazebo environment, we ran a second simulation to check if small changes in the initial conditions of the monocular-vision algorithm influence the final value of the scale factor. In this case, the house on the right is closer to the quadrotor, as shown in Figure 13. At the end of the simulation, the calculated scale factor values were $k_x = 0.232$ and $k_y = 0.166$, proving that the scale factor is different even for the same environment but with minor changes in the initial conditions.

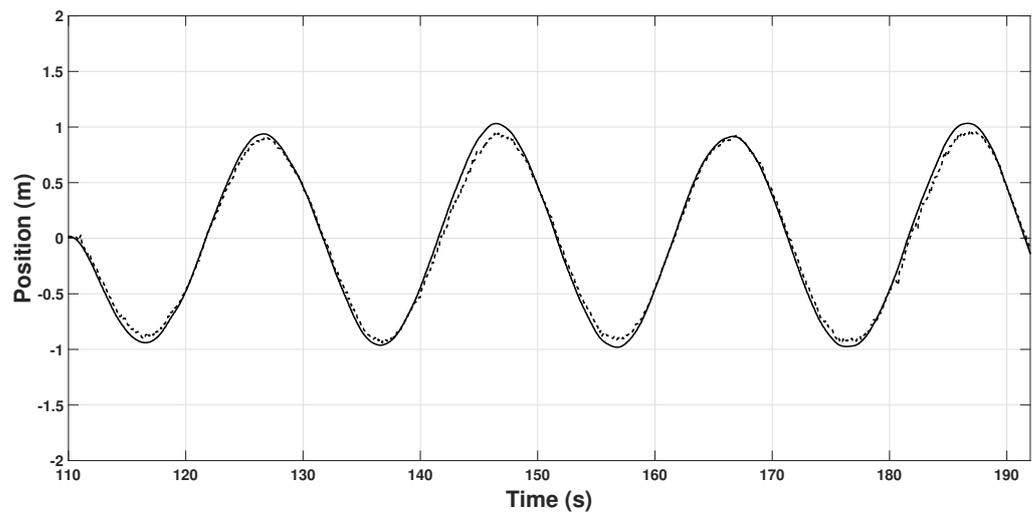


Figure 11. Quadrotor position on the Y^i axis. y (continuous line), $\frac{y_s}{k_y}$ (dashed line).

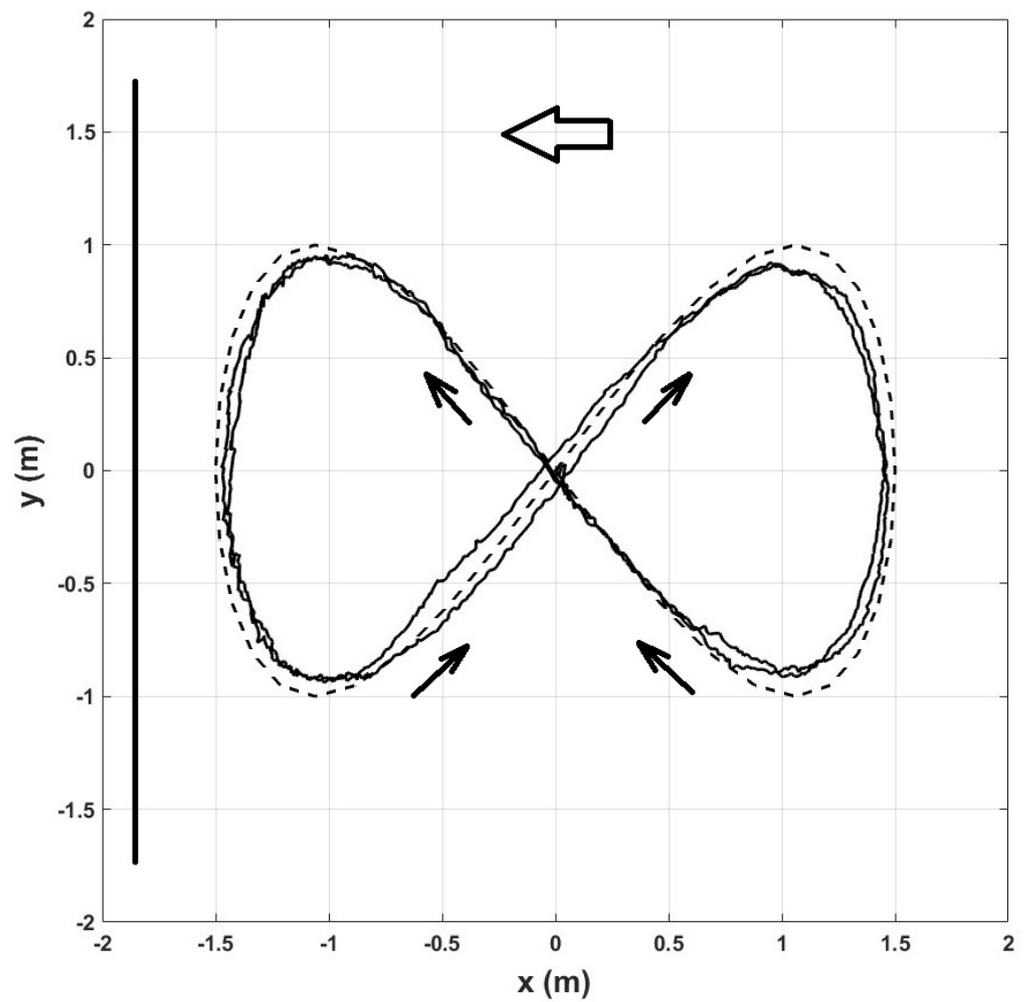


Figure 12. Lemniscate trajectory tracking using the scale factor estimator. Real position (continuous line), desired position (dashed line).



Figure 13. Gazebo environment with different initial conditions.

5. Conclusions

This article proposed a velocity observer in cascade with a scale factor estimator. The significant contributions of this work are listed next:

- The velocity observer does not neglect the Coriolis term, offering greater accuracy in fast flights.
- The scale factor estimator allows taking advantage of all the benefits of monocular cameras, obtaining the accuracy of a stereoscopic camera without increasing the processing power.
- Lyapunov's arguments prove asymptotic convergence to zero of the observer and estimator errors, and the simulations validate the correct performance and use of the proposed theory.
- It is illustrated that the scale factor is not the same in all axes, as some authors assume. It is even different from experimenting in the same environment if the initial conditions change.
- The proposed approach allows for position trajectory tracking to be performed directly using the measurements of a monocular-vision positioning algorithm, removing the limitations of a GPS or a motion capture system.

In future work, experiments will be carried out by a real quadrotor in more complex environments, combined with other kinds of computer vision algorithms such as person recognition or obstacle avoidance.

Author Contributions: Conceptualization, A.G.-C. and H.R.-C.; methodology, A.G.-C. and H.R.-C.; software, A.G.-C.; validation, A.G.-C.; formal analysis, H.R.-C.; investigation, A.G.-C. and H.R.-C.; writing—original draft preparation, A.G.-C.; writing—review and editing, H.R.-C.; visualization, A.G.-C.; supervision, H.R.-C.; project administration, H.R.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The simulation files are available upon request to the first author.

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

Appendix A. Proof of Theorem 1

Proof. The distance of the trajectories of systems (13) and (14) to the set \mathcal{M} is characterized by

$$\tilde{\eta} = \eta - \beta(\hat{\eta}, y)$$

The $\tilde{\eta}$ dynamic is given by

$$\dot{\tilde{\eta}} = f_1(\eta, y) - \frac{\partial \beta}{\partial \hat{\eta}} \varphi(\hat{\eta}, y) - \frac{\partial \beta}{\partial y} f_2(\eta, y) \quad (\text{A1})$$

thus, replacing $\varphi(\hat{\eta}, y)$ from Equation (16) into (A1), it follows that the $\tilde{\eta}$ dynamic becomes equal to (15). Finally, assumption A2 implies that (globally)

$$\lim_{t \rightarrow \infty} \beta(\hat{\eta}, y) = \eta \quad (\text{A2})$$

□

Appendix B. Proof of Proposition 1

Proof. The proof is divided into the following steps. In the first step, it is shown that the velocity observer error converges exponentially to zero. Then, it is proven that the trajectories of the estimation error do not explode in finite time. Finally, it is verified that the interconnection term between the observer and estimator dynamics satisfies a linear growth condition to the estimator error.

Note that considering Equation (27), the time derivative of the Lyapunov function (26) can be written as follows

$$\dot{v}_V = -\tilde{V}^b \Gamma_v \tilde{H} \tilde{V}^b \quad (\text{A3})$$

as a result, one has \tilde{V}^b converges exponentially to zero.

Now, to prove that the estimation error trajectories do not explode in finite time, consider the following Lyapunov function

$$v_K = \frac{1}{2} \tilde{K}^\top \tilde{K} \quad (\text{A4})$$

whose derivative with respect to time along the estimator dynamic (46) is

$$\begin{aligned} \dot{v}_K &= -\tilde{K}^\top \Gamma_k \left(\text{diag}(V^i) \text{diag}(V^i) + \text{diag}(\tilde{V}^i) \text{diag}(\tilde{V}^i) \right) \tilde{K} \\ &\quad + 2\tilde{K}^\top \Gamma_k \text{diag}(V^i) \text{diag}(\tilde{V}^i) \tilde{K} \\ &\quad - \tilde{K}^\top \Gamma_k \left(\text{diag}(V^i) - \text{diag}(\tilde{V}^i) \right) \text{diag}(\tilde{V}^i) K \\ &\quad + \tilde{K}^\top \Gamma_k \text{diag}(y_1) \Gamma_v \tilde{H} \tilde{V}^i \end{aligned} \quad (\text{A5})$$

thus, ($\lambda_M(\cdot)$ stands for the maximal eigenvalue of matrix \cdot .)

$$\begin{aligned} \dot{v}_K &\leq \lambda_M(\Gamma_k) \left(\|\text{diag}(V^i)\|^2 + \|\text{diag}(\tilde{V}^i)\|^2 \right) \|\tilde{K}\|^2 \\ &\quad + 2\lambda_M(\Gamma_k) \|\text{diag}(V^i)\| \|\text{diag}(\tilde{V}^i)\| \|\tilde{K}\|^2 \\ &\quad + \lambda_M(\Gamma_k) \left(\|\text{diag}(V^i)\| + \|\text{diag}(\tilde{V}^i)\| \right) \|\text{diag}(\tilde{V}^i)\| \|K\| \|\tilde{K}\| \\ &\quad + \lambda_M(\Gamma_k) \|\text{diag}(y_1)\| \lambda_M(\Gamma_v) \lambda_M(\tilde{H}) \|\tilde{V}^i\| \|\tilde{K}\| \end{aligned} \quad (\text{A6})$$

Hence, since the quadrotor dynamics is forward complete, the relationship in (A6) implies that the dynamic of \tilde{K} does not blow up in finite time. Note that if the error signal \tilde{V}^b is equal to zero the estimation error dynamic (46) reduces to

$$\dot{\tilde{K}} = -\Gamma_k \text{diag}(V^i) \text{diag}(V^i) \tilde{K} \quad (\text{A7})$$

and the time derivative of the Lyapunov function (A4) along the trajectories of (A7) becomes

$$\dot{v}_K = -\tilde{K}^\top \Gamma_k \text{diag}(V^i) \text{diag}(V^i) \tilde{K} \quad (\text{A8})$$

then, from Assumption 1 and selecting Γ_k as a positive-definite matrix, it follows that the estimation error \tilde{K} converges asymptotically to zero.

The last step in this proof is to prove that the interconnection term between the estimator and observer dynamics given by

$$\begin{aligned} \Psi &= -\Gamma_k \text{diag}(\tilde{V}^i) \text{diag}(\tilde{V}^i) \tilde{K} + 2\Gamma_k \text{diag}(V^i) \text{diag}(\tilde{V}^i) \tilde{K} \\ &\quad - \Gamma_k \left(\text{diag}(V^i) - \text{diag}(\tilde{V}^i) \right) \text{diag}(\tilde{V}^i) K \\ &\quad + \Gamma_k \text{diag}(y_1) \Gamma_v \tilde{H} \tilde{V}^i \end{aligned} \quad (\text{A9})$$

grows linearly with respect to \tilde{K} . From, Assumption 2 one has

$$\begin{aligned} \Psi &\leq \lambda_M(\Gamma_k) \left(\|\tilde{V}^i\|^2 + 2\kappa_0 \|\tilde{V}^i\| \right) \|\tilde{K}\| + \lambda_M(\Gamma_k) \left(\kappa_0 + \|\tilde{V}^i\| \right) \|\tilde{V}^i\| \|K\| \\ &\quad + \lambda_M(\Gamma_k) \kappa_1 \lambda_M(\Gamma_v) \lambda_M(\tilde{H}) \|\tilde{V}^i\| \end{aligned} \quad (\text{A10})$$

The inequality (A10) shows that the interconnection term Ψ grows linearly with respect to the estimation error \tilde{K} ; thus, Assumption 4.5 of [31] is satisfied. As a result, the cascade system (46) and (24) is asymptotically stable. Hence, the proof is concluded. \square

References

1. Gupte, S.; Mohandas, P.I.T.; Conrad, J.M. A survey of quadrotor Unmanned Aerial Vehicles. In Proceedings of the 2012 Proceedings of IEEE Southeastcon, Orlando, FL, USA, 15–18 March 2012; pp. 1–6.
2. Ganesan, R.; Raajini, X.M.; Nayyar, A.; Sanjeevikumar, P.; Hossain, E.; Ertas, A.H. BOLD: Bio-Inspired Optimized Leader Election for Multiple Drones. *Sensors* **2020**, *20*, 3134. [[CrossRef](#)] [[PubMed](#)]
3. Balamurugan, G.; Valarmathi, J.; Naidu, V.P.S. Survey on UAV navigation in GPS denied environments. In Proceedings of the 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, India, 3–5 October 2016; pp. 198–204.
4. Xie, N.; Lin, X.; Yu, Y. Position estimation and control for quadrotor using optical flow and GPS sensors. In Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 11–13 November 2016; pp. 181–186.
5. Arreola, L.; Montes de Oca, A.; Flores, A.; Sanchez, J.; Flores, G. Improvement in the UAV position estimation with low-cost GPS, INS and vision-based system: Application to a quadrotor UAV. In Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS), Dallas, TX, USA 12–15 June 2018; pp. 1248–1254.
6. Scaramuzza, D.; Fraundorfer, F. Visual Odometry [Tutorial]. *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92. [[CrossRef](#)]
7. Anderson, M.L.; Brink, K.M.; Willis, A.R. Real-Time Visual Odometry Covariance Estimation for Unmanned Air Vehicle Navigation. *J. Guid. Control. Dyn.* **2019**, *42*, 1272–1288. [[CrossRef](#)]
8. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
9. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Computer Vision—ECCV 2014*; Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 834–849.
10. Chen, Y.; Zhou, Y.; Lv, Q.; Deveerasetty, K.K. A Review of V-SLAM. In Proceedings of the 2018 IEEE International Conference on Information and Automation (ICIA), Wuyishan, China, 11–13 August 2018; pp. 603–608.
11. Filipenko, M.; Afanasyev, I. Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment. In Proceedings of the 2018 International Conference on Intelligent Systems (IS), Funchal, Portugal, 25–27 September 2018; pp. 400–407.
12. Mouaad, B.; Razika, B.Z.; Ramzi, R.H.; Karim, C. Control Design and Visual Autonomous Navigation of Quadrotor. In Proceedings of the 2019 International Conference on Advanced Electrical Engineering (ICAEE), Algiers, Algeria, 19–21 November 2019; pp. 1–7.
13. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.
14. Lin, Y.; Wang, J.; Shi, Z.; Zhong, Y. Reinitializable and scale-consistent visual navigation for UAVs. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 26–28 July 2017; pp. 5871–5876.
15. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [[CrossRef](#)]

16. Engel, J.; Stücker, J.; Cremers, D. Large-scale direct SLAM with stereo cameras. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1935–1942.
17. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Trans. Robot.* **2017**, *33*, 249–265. [[CrossRef](#)]
18. Giubilato, R.; Pertile, M.; Debei, S. A comparison of monocular and stereo visual FastSLAM implementations. In Proceedings of the 2016 IEEE Metrology for Aerospace (MetroAeroSpace), Florence, Italy, 22–23 June 2016. doi: 10.1109/MetroAeroSpace.2016.7573217. [[CrossRef](#)]
19. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2003.
20. Esrafilian, O.; Taghirad, H.D. Autonomous flight and obstacle avoidance of a quadrotor by monocular SLAM. In Proceedings of the 2016 4th International Conference on Robotics and Mechatronics (ICROM), Tehran, Iran, 26–28 October 2016; pp. 240–245.
21. 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](#)] [[PubMed](#)]
22. Ludhiyani, M.; Rustagi, V.; Dasgupta, R.; Sinha, A. Multirotor dynamics based online scale estimation for monocular SLAM. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 6475–6481.
23. Lee, S.H.; de Croon, G. Stability-Based Scale Estimation for Monocular SLAM. *IEEE Robot. Autom. Lett.* **2018**, *3*, 780–787. [[CrossRef](#)]
24. Astolfi, A.; Karagiannis, D.; Ortega, R. *Nonlinear and Adaptive Control with Applications*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 187.
25. Nieto-Hernández, L.; Gómez-Casasola, A.A.; Rodríguez-Cortés, H. Monocular SLAM Position Scale Estimation for Quadrotor Autonomous Navigation. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 1359–1364.
26. Gómez-Casasola, A.; Rodríguez-Cortés, H. Sensor Fusion for Quadrotor Autonomous Navigation. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 5219–5224.
27. Leishman, R.C.; Macdonald, J.C.; Beard, R.W.; McLain, T.W. Quadrotors and Accelerometers: State Estimation with an Improved Dynamic Model. *IEEE Control Syst. Mag.* **2014**, *34*, 28–41. [[CrossRef](#)]
28. Rodríguez-Cortés, H. Aportaciones al control de vehículos aéreos no tripulados en México. *Rev. Iberoam. Autom. Inform. Ind.* **2022**, *19*, 430–441. [[CrossRef](#)]
29. Bramwell, A.R.S.; Done, G.; Balmford, D. *Bramwell's Helicopter Dynamics*, 2nd ed.; Butterworth-Heinemann: Oxford, UK, 2001.
30. Monajjemi, M. Ardrone_autonomy: A ROS Driver for AR-Drone 1.0 & 2.0. 2012. Available online: https://github.com/AutonomyLab/ardrone_autonomy (accessed on 10 September 2022).
31. Sepulchre, R.; Jankovic, M.; Kokotovic, P.V. *Constructive Nonlinear Control*; Springer: London, UK, 2012.