sensors

MDPI

Article

A Sensor Fusion Approach to Observe Quadrotor Velocity

José Ramón Meza-Ibarra [1,†], Joaquín Martínez-Ulloa [1,†], Luis Alfonso Moreno-Pacheco [1,*,†] and Hugo Rodríguez-Cortés [2,†,‡]

[1] Sección de Estudios de Posgrado, Escuela Superior de Ingeniería Mecánica y Eléctrica, Instituto Politécnico Nacional, Av. Instituto Politécnico Nacional, S/N, Col. Lindavista, Ciudad de México 07738, Mexico; jmezai1600@alumno.ipn.mx (J.R.M.-I.); jmartinezu1901@alumno.ipn.mx (J.M.-U.)

[2] Departamento de Ingeniería Eléctrica y Electrónica, Instituto Tecnológico Autónomo de México, Río Hondo 1, Tizapán, Mexico City 01080, Mexico; hugo.rodriguez@itam.mx or hrodriguez@cinvestav.mx

[*] Correspondence: lamoreno@ipn.mx

[†] All authors contributed equally to this work.

[‡] On sabbatical leave from Departamento de Ingeniería Eléctrica, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Av. Instituto Politécnico Nacional No. 2508, Col. San Pedro Zacatenco, Ciudad de México 07360, Mexico.

Abstract: The growing use of Unmanned Aerial Vehicles (UAVs) raises the need to improve their autonomous navigation capabilities. Visual odometry allows for dispensing positioning systems, such as GPS, especially on indoor flights. This paper reports an effort toward UAV autonomous navigation by proposing a translational velocity observer based on inertial and visual measurements for a quadrotor. The proposed observer complementarily fuses available measurements from different domains and is synthesized following the Immersion and Invariance observer design technique. A formal Lyapunov-based observer error convergence to zero is provided. The proposed observer algorithm is evaluated using numerical simulations in the Parrot Mambo Minidrone App from Simulink-Matlab.

Keywords: visual odometry; optical flow; sensor fusion; observer design; velocity observation

check for updates

Citation: Meza-Ibarra, J.R; Martínez-Ulloa, J.; Moreno-Pacheco, L.A.; Rodríguez-Cortés, H. A Sensor Fusion Approach to Observe Quadrotor Velocity. *Sensors* **2024**, *24*, 3605. https://doi.org/10.3390/s24113605

Academic Editor: Andrzej Stateczny

Received: 24 April 2024
Revised: 24 May 2024
Accepted: 30 May 2024
Published: 3 June 2024

1. Introduction

Thanks to the advancement of unmanned aerial vehicles, these have gained ground in various applications, including law enforcement, precision agriculture, and architectural/industrial inspection. Additionally, researchers have chosen to use this aircraft type, which offers excellent academic advantages, to test new nonlinear control theory methods. Crewless aircraft have made a quantum leap, and the nonlinear control theory has evolved. The basis of the classical nonlinear control theory, built by, for instance, [1,2], has been adapted rapidly into this century's new technologies, as reported in [3,4]. Hence, quadrotor aerial vehicles have become a worldwide standard platform for robotics research. In particular, the Parrot Mambo Minidrone App in Simulink-Matlab offers excellent functionality for testing new control and state observation algorithms. It provides semi-realistic inertial and visual sensors, making real-time implementation straightforward. In reference to the first developments of sensor fusion for unmanned aerial vehicles, we can find that in [5], the measurements from an Inertial Navigation System (INS) and GPS sensors are fused by using a Kalman filter. The information for the simulation experiment was used to examine the data of both sensors; tool implementations of the filter were considered. The research article [6] establishes a basic requirement for an autonomous robot merging the data of different sensors; odometric and sonar sensors are fused together by means of an Extended Kalman Filter (EKF). The adaptive algorithm performs a precise localization of the vehicle and can be established in a wide range of experimental situations. The researchers in [7] describe a simple yet powerful statistical technique for fusing information from different sensors that aims to obtain the spectral error densities of the navigation gravity model, completing a robust aerial model that absorbs vertical disturbances. The information is computed in the time-dependent frequency.

In [8], the authors present a real-time implementation of position controllers based on the Adaptive–Proportional–Integral Derivative (APID) method for the Parrot Mambo Minidrone. An adaptive mechanism based on a second-order sliding mode control is also included to modify the conventional parameters of the altitude controller. Simulations and experimental flights validate the success of this approach. The research reported in [9] synthesizes Proportional–Integral–Derivative (PID), Linear Quadratic Regulator (LQR), and Model Predictive Control (MPC) algorithms for a Parrot Minidrone, and the control algorithms' strengths and weaknesses are analyzed. These control techniques were chosen considering that the MPC design for Parrot Minidrones is unavailable in the existing literature. The automatic code generation capabilities offered by the Simulink coder facilitated the experimental implementation of the proposed control algorithms. The use of a quadrotor linear model evidently causes a relatively large mismatch between simulations and experimental results. The work in [10] presented the real-time implementation of a novel Cartesian translational robust control strategy for the Parrot Mambo Minidrone. Multiple data were collected from actual flight experiments involving a set of four Parrot Mambo Minidrones. A first-order dynamic with time delay was identified as the mathematical model for the Cartesian translational dynamic. The control strategy was designed based on a parameter variant discrete-time linear system. Reference [11] reports the design and implementation of a cascade attitude controller for the Parrot Mambo Minidrone. The controller was designed using the triple-step and nonlinear integral sliding mode control NISMC method considering the three-DOF nonlinear model. In [12], a hand gesture-based drone controller for the Parrot Mambo Minidrone is developed; the controller can perform take-off, hovering, and landing operations. The delay between executing each command is only three seconds, and the system accuracy obtained for hand gesture detection is remarkably good. The system can be further extended by modifying the controller so that the UAV can move and perform multiple commands using hand gestures.

Current directions aim to develop nonlinear observer strategies for autonomous navigation using only onboard sensors. The work in [13] proposed an embedded fast Nonlinear Predictive Model (NMPC) algorithm. This controller ensures a stable and safe flight of micro aerial robots relying solely on onboard sensors to localize themselves. The implemented controller can drive the micro aerial vehicle to track a path and outstand external disturbances. The design of a computationally efficient optical flow algorithm for tiny multirotor aerial vehicles, also known as pocket drones, is the focus of the work reported [14]. The Edge-Flow algorithm uses a compressed representation of an image frame to match it with the one in the previous time step; the adaptive time horizon also enables it to detect sub-pixel flow, from which velocities can be observed. Reference [15] presents a fusion filter design using the Kalman Filter (KF) complemented with the No Motion No Integration Filter (NMNI) to reduce noise influence to observe the tilt angle using only one accelerometer and one gyroscope. This approach relieves the burden of multiple sensors for attitude tracking, reducing battery energy consumption and helping the drone obtain the precise angle under rotor vibrations.

Translational velocity observation for quadrotors is an appealing problem from practical and theoretical perspectives. In [16], a globally exponential convergent observer based on nonlinear adaptive techniques is proposed. The observer uses measurements from an Attitude and Heading Reference System (AHRS). Numerical simulations evaluate the observer algorithm that shows good performance in the presence of noisy acceleration measurements. Reference [17] reports the design of a deterministic quadrotor translational velocity observer employed to reconstruct the scale factor of the position determined by a Simultaneous Localization And Mapping (SLAM) algorithm. The translational velocity observer uses the translational acceleration and the non-scaled translational position; its performance is evaluated through numerical simulations. Translational velocity observation is also an essential issue for fixed-wing aircrafts. Reference [18] presents an exponentially stable nonlinear wind velocity observer for fixed-wing unmanned aerial vehicles. The proposed observer employs a GNSS-aided Inertial Navigation System (INS), an attitude observer, and a pitot-static probe measuring dynamic pressure and airspeed in the longitudinal direction. The proposed observer estimates wind velocity and, as by-products, the angles of attack, sideslip, and scaling factor of the pitot-static probe measurement without requiring UAV maneuvers with the Persistence of Excitation (PE). The algorithm is well-suited for embedded systems and was tested through simulations. Finally, in [19], a uniform, semi-global, exponentially stable nonlinear observer for attitude, gyro bias, position, velocity, and specific force estimation for a fixed-wing UAV is presented. The nonlinear observer uses inertial and visual measurements without any assumptions about the flight altitude or the structure of the terrain being recorded. Experimental data from a UAV test flight and simulated data show that the nonlinear observer performs robustly.

Examining the existing literature reveals a significant gap in quadrotor translational velocity observations. No deterministic observer complements inertial and visual measurements with a formal proof of convergence of the observation error and a semi-realistic numerical simulation. By incorporating novel methodologies such as the Immersion and Invariance method, this research fills these gaps and presents a technique to complementarily fuse both measurements, accompanied by a formal proof of the convergence to zero of the observation error. This research was driven by an aim to enhance quadrotor capabilities to fly autonomously using only onboard sensors. In relation to the autonomous performance of aerial vehicles, we can find in [20], mounted-based stations, which are expected to become an integral component of future intelligent transportation systems. The aerial vehicle, integrated with the ground vehicular network, is used to bridge coverage gaps, offer broader communication services, and improve the network connection stability. Its self understanding is the main task, which is performed by the autonomous unmanned vehicle. As mentioned, a fundamental goal of the autonomous UAV is the development of navigation systems that help to realize any kind of aerial operations; inside [21], a nano-UAV learns to detect and fly through programmed trajectories with an autonomous navigation system based on neural networks. This research article is extremely pretentious because, besides the autonomous requirements, it is needed to establish appropriate sensor fusion, as well as the overall perspective of the whole aerial environment. Key insights from this research include the complementary nature of the inertial and visual sensors. Moreover, the findings reported in this article complement the work presented in [17]. The nonlinear translational velocity observer of [17] degrades when the quadrotor remains in hover; this problem is overcome by adding the optical flow measurement. Adding the optical measurement also gives the proposed observer an advantage over the observer reported in [16]. Finally, it is important to observe that the nonlinear observers reported in [18,19] use different sensors.

This paper introduces a novel sensor fusion strategy for observing the quadrotor translational velocity. The proposed observer algorithm uniquely fuses inertial and visual measurements. This fusion strategy is synthesized following the Immersion and Invariance approach method introduced in [3]. The observer error convergence to zero is guaranteed using Lyapunov theory arguments, and its performance is evaluated through numerical simulations in the semi-realistic Parrot Minidrone app from Matlab-Simulink. It is demonstrated that the proposed observer operates effectively under noisy measurements and considering different quadrotor motions.

This work has the following structure: Section 2 presents the quadrotor model and describes the available measurements. Section 3 reports the sensor fusion algorithm and formally states the nonlinear observer design. Section 4 is devoted to the numerical simulation study, the observation error, and the stability properties of the designed observer; finally, Section 5 presents the conclusions of this work.

2. Quadrotor Mathematical Model and Available Measurements

2.1. Quadrotor Dynamics

The basic structure of the dynamic quadrotor model has been reported in various research papers and books. This work considers the dynamic quadrotor model reported in [17,22]. The model contains states expressed in two frames of reference, the inertial and the fixed body, and is described by the following set of differential equations. See Figure 1.

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

where $X = \begin{bmatrix} x & y & z \end{bmatrix}^\top$ is the quarotor position in the inertial reference frame. $R \in SO(3)$ is the rotation matrix from the fixed body to inertial coordinates with

$$SO(3) = \left\{ R \in \mathbb{R}^{3\times3} \,\middle|\, R^\top R = I, \det(R) = 1 \right\}$$

where $I \in \mathbb{R}^{3\times3}$ is the identity matrix, and $V^b = \begin{bmatrix} u & v & w \end{bmatrix}^\top$ denotes the translational velocity expressed in the fixed body frame. Moreover, m represents the vehicle's mass, g is the gravitational force constant, $e_3 = [0 \quad 0 \quad 1]^\top$, T_T is the total thrust force produced by the four rotors,

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and $S(\cdot)$ is a skew symmetric matrix such that

$$a \times b = S(a)b, \quad \forall\, a, b \in \mathbb{R}^3,$$

$J = \text{diag}\{J_{xx}, J_{yy}, J_{zz}\}$ is the inertia matrix, $\Omega = \begin{bmatrix} p & q & r \end{bmatrix}^\top$ is the quadrotor rotational velocity expressed in the fixed-body frame coordinates, and M^b is the vector of moments generated by the differential rotor's thrust and rotor's reaction moment.

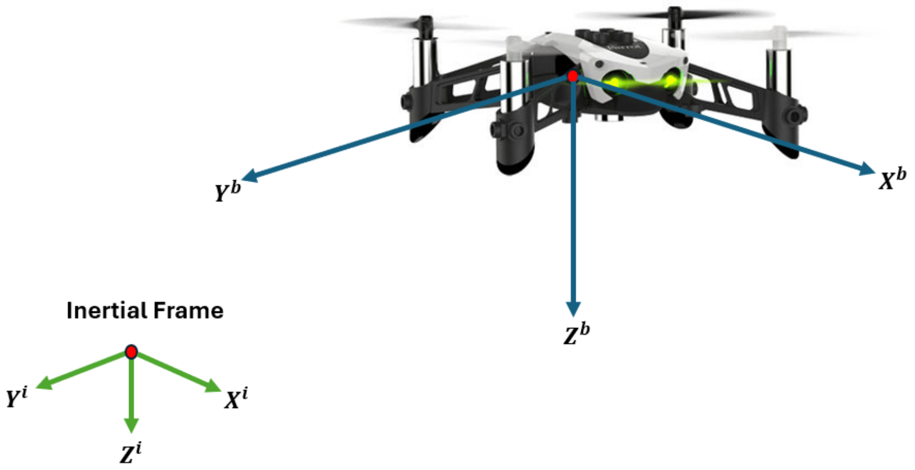

Figure 1. Mambo Parrot's coordinate frames.

2.2. Available Measurements

This work considers that the quadrotor has an Inertial Measurement Unit (IMU), a monocular camera pointing downwards, and a height sensor. Thus, the following signals are available.

2.2.1. Quadrotor's Specific Translational Acceleration

As the leading electronic sensor, multi-rotors carry an IMU, which provides information on the fixed body coordinates. An IMU delivers the specific translational acceleration a^b, the angular velocity Ω, and the intensity of the Earth's magnetic field. According to the work reported in [23], the specific translational acceleration measured by the IMU's accelerometer mounted on board an aerial vehicle is given by

$$a^b = \frac{1}{m} F_T^b - g R^\top e_3 \tag{2}$$

where F_T^b models the total external forces acting on the vehicle where the IMU is mounted. From the second equation in (1), it follows that

$$F_T^b = mgR^T e_3 - T_T e_3 - \mu H V^b \tag{3}$$

Notice that the above equation does not include the term $-mS(\Omega)V^b$ because the Coriolis acceleration is an inertial force. Substituting (3) into (2), the specific acceleration force measured by an accelerometer onboard a quadrotor is

$$a^b = -\frac{T_T}{m} e_3 - \frac{\mu}{m} H V^b \tag{4}$$

The first available measurement is the specific translational acceleration, which element-to-element reads in the following way,

$$y_1 = a^b = \begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} = -\frac{1}{m} \begin{bmatrix} \mu u \\ \mu v \\ T_T \end{bmatrix} \tag{5}$$

2.2.2. Quadrotor's Attitude and Angular Velocity

The IMU data processed through an estimation algorithm constitutes an Attitude and Heading Reference System (AHRS) that provides the quadrotor attitude R and angular velocity Ω. The AHRS can deliver the attitude using a parameterization such as Euler angles or quaternions or directly delivering the rotation matrix R. Thus, one has

$$\begin{array}{rcl} y_2 & = & R \\ y_3 & = & \Omega \end{array} \tag{6}$$

2.2.3. Optical Flow

Nowadays, the second crucial sensor for autonomous navigation is a monocular camera. The pixel velocity can be measured and related to the quadrotor velocity by processing the camera image.

The cornerstone to interpreting an image inside a computer is brightness. The camera sensors integrate the irradiance from the scene so that $I(x_p, y_p)$ defines the brightness of the pixel located at (x_p, y_p); thus, the k image captured at time t can be characterized as [24],

$$I_k(x_p, y_p, t) : \mathcal{R} \subset \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}_+$$

Computer vision deals with extracting meaningful information from images; this is extracting meaningful information from brightness. A much more straightforward problem is computing image motion information from brightness. Consider two images of the same pixel location, $I_1(x_p, y_p, t_1)$ and $I_2(x_p, y_p, t_2)$, taken from infinitesimally close vantage points and at infinitesimally consecutive time instants $t_2 \approx t_1$; thus,

$$\begin{array}{rcl} I_1(x_p(t_1), y_p(t_1), t_1) & = & I_1(x_p(t_1), y_p(t_1), t_1) \\ I_2(x_p(t_2), y_p(t_2), t_2) & = & I_2(x_p(t_1 + \Delta t), y_p(t_1 + \Delta t), t_1 + \Delta t) \end{array}$$

where $t_2 = t_1 + \Delta t$ with Δt an infinitesimal time increment. Assume that only pixels belonging to flat and parallel image plane portions and moving parallel to the image plane are considered. Then,

$$\begin{aligned} x_p(t_1 + \Delta t) &= x_p(t_1) + u_p \Delta t \\ y_p(t_1 + \Delta t) &= y_p(t_1) + v_p \Delta t \end{aligned}$$

with u_p and v_p the pixel velocity. As a result,

$$I_1(x_p(t_1), y_p(t_1), t_1) = I_1(x_p(t_1) + u_p\Delta t, y_p(t_1) + v_p\Delta t, t_1 + \Delta t) \tag{7}$$

Expanding the right-hand-side of Equation (7) by the Taylor series, one has

$$I_1(x_p(t_1), y_p(t_1), t_1) = I_1(x_p(t_1), y_p(t_1), t_1) + \frac{\partial I_1}{\partial x_p} u_p \Delta t + \frac{\partial I_1}{\partial y_p} v_p \Delta t + \frac{\partial I_1}{\partial t} \Delta t + O(\Delta t^2) \tag{8}$$

Neglecting the high-order terms, it follows that

$$\frac{\partial I_1}{\partial x_p} u_p + \frac{\partial I_1}{\partial y_p} v_p + \frac{\partial I_1}{\partial t} = 0 \tag{9}$$

this equation is known as the brightness constancy constraint [24] or the optical flow constraint equation [25,26]. Note that it is impossible to compute the speeds u_p and v_p perpendicular to the image gradient using the brightness constancy constraint; this drawback is known as the aperture problem [27]. It is necessary to evaluate the brightness constancy constraint in each pixel location belonging to a region where u_p and v_p can be assumed constant, for example, the window

$$\begin{aligned} W_i(x_p, y_p) = & \Big\{ (x_p, y_p) \in \mathcal{R}_W \subset \mathcal{R}, \delta \in \mathbb{R}_+ \mid |x_p - x_p^i| + |y_p - y_p^i| \leq \delta, \\ & \text{for some } (x_p^i, y_p^i) \in \mathcal{R}_W \Big\} \end{aligned}$$

with $\mathcal{R}$ the image plane. If the window $W_i(x_p, y_p)$ is fixed inside the image plane, the computed speeds u_p and v_p are known as the optical flow. Using the differential method proposed by Lucas–Kanade [28], the computation of constant values of u_p and v_p in each small neighborhood $W_i(x_p, y_p)$ can be implemented as the minimization of

$$\sum_{(x_p, y_p) \in W} Q(x_p, y_p)^2 \left[\frac{\partial I_1}{\partial x_p} u_p + \frac{\partial I_1}{\partial y_p} v_p + \frac{\partial I_1}{\partial t} \right]^2$$

with $Q(x)$, a function giving more influence to constraints at the center of $W_i(x_p, y_p)$ than those on the boundary. As reported in [29], this method is one of the most reliable.

Consider a camera onboard the aerial vehicle monitoring several characteristic points located at $P_i = \begin{bmatrix} x_c^i & y_c^i & z_c^i \end{bmatrix}^\top$, as illustrated in Figure 2. It is worth mentioning that this Figure comes from [30] with slight additions. The velocity of each point relative to the camera is given by:

$$\dot{P}_i = -y_3 \times P_i - V^b \tag{10}$$

From the perspective projection condition, it follows that the location of each characteristic point projects on the image plane as follows

$$\begin{aligned} x_p^i &= f \frac{x_c^i}{z_c^i} \\ y_p^i &= f \frac{y_c^i}{z_c^i} \end{aligned} \tag{11}$$

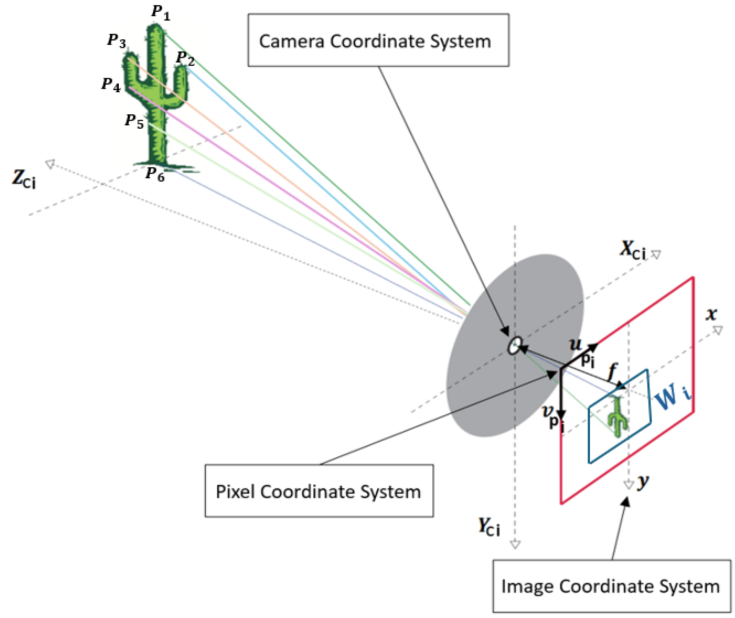

Figure 2. Pinhole camera principle [30].

with f the focal length of the camera lens. Combining (10) and (11), one obtains

$$\begin{bmatrix} u_p^i \\ v_p^i \end{bmatrix} = \begin{bmatrix} -\frac{f}{z}u + \frac{x_p^i}{z}w + \frac{x_p^i y_p^i}{f}p - \left(f + \frac{{x_p^i}^2}{f}\right)q + y_p^i r \\ -\frac{f}{z}v + \frac{y_p^i}{z}w + \left(f + \frac{{y_p^i}^2}{f}\right)p - \frac{x_p^i y_p^i}{f}q - x_p^i r \end{bmatrix} \tag{12}$$

the pixel velocity (u_p^i, v_p^i) registered by the camera due to the quadrotor motion.

Figure 2 shows that $z_c = z$. Using the differential method proposed by Lukas-Kanade to determine u_p and v_p in a region $W_i(x_p, y_p)$, it is possible to obtain the quadrotor translational velocity as follows

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\frac{z}{f}u_p + \frac{x_p}{f}w + z\frac{x_p y_p}{f^2}p - \frac{z}{f}\left(f + \frac{x_p^2}{f}\right)q + z\frac{y_p}{f}r \\ -\frac{z}{f}v + \frac{y_p}{f}w + \frac{z}{f}\left(f + \frac{y_p^2}{f}\right)p - z\frac{x_p y_p}{f^2}q - z\frac{x_p}{f}r \end{bmatrix} \tag{13}$$

in an image region $W_i(x_p, y_p)$ for some i.

Consider the following assumption,

Assumption 1. *The optical flow is computed in a region W_i that contains the pixel image origin; this is, $x_p = y_p = 0$. Moreover, the quadrotor has a height controller that ensures $z \approx \bar{z}$ for some constant $\bar{z}$ and $w \approx 0$.*

Under Assumption 1, the Equation (13) becomes

$$y_4 = \begin{bmatrix} u_p \\ v_p \end{bmatrix} = -H_1 V^b + H_2 y_3 \tag{14}$$

with

$$H_1 = \begin{bmatrix} \frac{f}{\bar{z}} & 0 & 0 \\ 0 & \frac{f}{\bar{z}} & 0 \end{bmatrix}, \; H_2 = \begin{bmatrix} 0 & -f & 0 \\ f & 0 & 0 \end{bmatrix}$$

a measurable signal.

2.2.4. Height Sensor

Ultrasonic or laser devices to measure distance can be mounted on quadrotors. Hence, it is assumed that the following signals are also measured

$$\begin{aligned} y_5 &= z \\ y_6 &= w \end{aligned} \tag{15}$$

thus, the matrix H_1 is also measurable since it can be expressed as

$$H_1 = \begin{bmatrix} \frac{f}{y_5} & 0 & 0 \\ 0 & \frac{f}{y_5} & 0 \end{bmatrix}$$

Finally, note that in terms of the measurable signals, the second equation in (7) can be expressed as

$$\dot{V}^b = g y_2^\top e_3 + y_1 - s(y_3) V^b \tag{16}$$

3. Nonlinear Observer Design

Sensor fusion has become essential for solving mobile robotics state observation problems [31]. Better spatial and temporal coverage, robustness to sensor failures, and increased state observation accuracy are the main desirable properties of sensor fusion algorithms. This research article proposes a sensor fusion algorithm to estimate the quadrotor Cartesian velocity based on the Immersion and Invariance observer design technique proposed in [3]. The proposed fusion algorithm can be classified as a complementary fusion across domains [32,33]. Both considered sensors measure the same quantity in different domains and in acceleration and pixel velocity, and they work in a complementary configuration. Figure 3 illustrates schematically the proposed sensor fusion. The sensor fusion algorithm is designed in the deterministic nonlinear time-invariant framework.

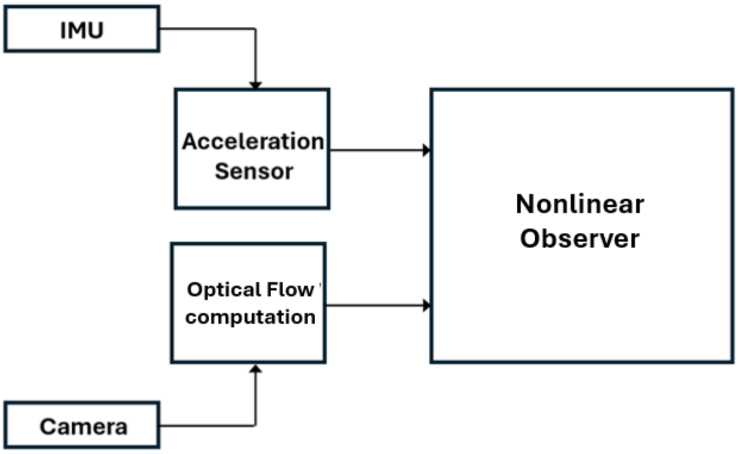

Figure 3. Sensor fusion.

The observer design method can be explained as follows. Consider the following non-linear, deterministic, time-invariant system [3].

$$\begin{aligned} \dot{\eta} &= f_1(\eta\,,\, \mathbf{y}) \\ \dot{\mathbf{y}} &= f_2(\eta\,,\, \mathbf{y}) \end{aligned} \tag{17}$$

where $\eta \in \mathcal{R} \subset \mathbb{R}^n$ and $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^m$ are the unmeasured and measured states, correspondingly.

Definition 1. *The dynamic system*

$$\dot{\hat{\eta}} = \phi(\hat{\eta}\,,\mathbf{y}) \tag{18}$$

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

$$\mathcal{M} = \{(\eta\,,\hat{\eta}\,,\mathbf{y}) \subset \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \,|\, \beta(\hat{\eta}\,,\mathbf{y}) = \eta\} \tag{19}$$

has the following properties,

- *$\mathcal{M}$ is positively invariant;*
- *All trajectories of (17) and (18) that start in a neighborhood of $\mathcal{M}$ asymptotically converge to $\mathcal{M}$.*

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

Theorem 1. *Consider the system (17). Assume that the vector fields $f_1(\eta\,,\mathbf{y})$ and $f_2(\eta\,,\mathbf{y})$ are forward complete and that there exist differentiable maps $\beta : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n$ such that*

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

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

A2 The dynamic system

$$\dot{\tilde{\eta}} = f_1(\tilde{\eta} + \beta(\hat{\eta}\,,\mathbf{y})\,,\mathbf{y}) - f_1(\beta(\hat{\eta}\,,\mathbf{y})\,,\mathbf{y}) - \frac{\partial\beta}{\partial\mathbf{y}}(f_2(\tilde{\eta} + \beta(\hat{\eta}\,,\mathbf{y})\,,\mathbf{y}) - f_2(\beta(\hat{\eta}\,,\mathbf{y})\,,\mathbf{y})) \tag{20}$$

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

Then, the system (18) with,

$$\phi(\hat{\eta}\,,\mathbf{y}) = \left(\frac{\partial\beta}{\partial\hat{\eta}}\right)^{-1}\left(f_1(\beta(\hat{\eta}\,,\mathbf{y})\,,\mathbf{y}) - \frac{\partial\beta}{\partial\mathbf{y}} f_2(\beta(\hat{\eta}\,,\mathbf{y})\,,\mathbf{y})\right) \tag{21}$$

is a (global) observer for (17).

Remark 1. *The result in Theorem 1 is a simplified version of the general observer design theory reported in [3]. The proof of Theorem 1 was reported in [17]. The sensor fusion characteristics of the observer in Definition 1 are as follows. Assume that two measurements $\mathbf{y}_1$ and $\mathbf{y}_2$ contain information on the nonmeasurable state η. Then, it is possible to define a function $\gamma(\mathbf{y}_1, \mathbf{y}_2)$ that fuses both measurements, and then the function $\beta(\hat{\eta}, \gamma(\mathbf{y}_1, \mathbf{y}_2))$ integrates the fusion to the observer design procedure.*

Cartesian Velocity Observer

Here, the quadrotor Cartesian velocity observer is designed. The observer fuses all measurements described in Section 2.2. First, using all available measurements, the following measurements are tailored

$$\begin{aligned} \bar{y}_1 &= \begin{bmatrix} a_x^b \\ a_y^b \\ y_6 \end{bmatrix} = \bar{H}_1 V^b \\ \bar{y}_4 &= \begin{bmatrix} u_p \\ v_p \\ y_6 \end{bmatrix} = \bar{H}_2 V^b + \bar{H}_3 y_3 \end{aligned} \tag{22}$$

with

$$\bar{H}_1 = \begin{bmatrix} -\frac{\mu}{m} & 0 & 0 \\ 0 & -\frac{\mu}{m} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \bar{H}_2 = \begin{bmatrix} -\frac{f}{\bar{y}_5} & 0 & 0 \\ 0 & -\frac{f}{\bar{y}_5} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \bar{H}_3 = \begin{bmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

From the definition of the manifold (19), the observer error is defined as

$$\tilde{V}^b = V^b - \beta_1(\hat{V}^b, \sigma) \tag{23}$$

with

$$\dot{\sigma} = \gamma(\bar{y}_1, \bar{y}_4) = \frac{1}{\alpha_1 + \alpha_2}(\alpha_1 \bar{y}_1 + \alpha_2 \bar{y}_4) \tag{24}$$

with α_1 and α_2 scalar constants. Note that the function $\gamma(\bar{y}_1, \bar{y}_4)$ fuses the information of the same quantity, translational velocity, expressed in different domains, body axes acceleration $\bar{y}_1$, and optical flow $\bar{y}_4$. The scalars α_1 and α_2 modulate the fusion. The time derivative of the observer error $\tilde{V}^b$ reads as

$$\dot{\tilde{V}}^b = \dot{V}^b - \frac{\partial \beta_1}{\partial \hat{V}^b}\dot{\hat{V}}^b - \frac{\partial \beta_1}{\partial \sigma}\gamma(\bar{y}_1, \bar{y}_4) \tag{25}$$

Using (16) and (22), one obtains

$$\dot{\tilde{V}}^b = g y_2^T e_3 + y_1 - S(y_3)V^b - \frac{\partial \beta_1}{\partial \hat{V}^b}\dot{\hat{V}}^b + \frac{1}{\alpha_1 + \alpha_2}\frac{\partial \beta_1}{\partial \sigma}\Big(\alpha_1 \bar{H}_1 V^b + \alpha_2(\bar{H}_2 V^b + \bar{H}_3 y_3)\Big) \tag{26}$$

Now, to express (26) in terms of the observer error, the Equation (23) is solved for V^b and substituted in (26). Thus,

$$\begin{aligned} \dot{\tilde{V}}^b \;=\;& g y_2^T e_3 + y_1 - S(y_3)\Big(\tilde{V}^b + \beta_1(\hat{V}^b, \sigma)\Big) - \tfrac{\partial \beta_1}{\partial \hat{V}^b}\dot{\hat{V}}^b \\ &+ \frac{1}{\alpha_1 + \alpha_2}\frac{\partial \beta_1}{\partial \sigma}\Big(\alpha_1 \bar{H}_1\Big(\tilde{V}^b + \beta_1(\hat{V}^b, \sigma)\Big) + \alpha_2(\bar{H}_2\Big(\tilde{V}^b + \beta_1(\hat{V}^b, \sigma)\Big) + \bar{H}_3 y_3)\Big) \end{aligned} \tag{27}$$

Defining

$$\beta_1(\hat{V}^b, \sigma) = \hat{V}^b + \Gamma\sigma \tag{28}$$

with Γ a matrix gain, the state observer dynamic can be defined as follows

$$\dot{\hat{V}}^b = g y_3^T e_3 + y_1 - S(y_3)\beta_1 + \frac{1}{\alpha_1 + \alpha_2}\Gamma\Big(\alpha_1 \bar{H}_1 \beta_1(\hat{V}^b, \sigma) + \alpha_2(\bar{H}_2 \beta_1(\hat{V}^b, \sigma) + \bar{H}_3 y_3)\Big) \tag{29}$$

It is important to verify that the state observer dynamic depends only on available measurements and known parameters. Then, the following vector differential equation described the observer error dynamic

$$\dot{\tilde{V}}^b \;=\; -S(y_3)\tilde{V}^b + \frac{1}{\alpha_1 + \alpha_2}\Gamma(\alpha_1 \bar{H}_1 + \alpha_2 \bar{H}_2)\tilde{V}^b \tag{30}$$

Hence, one has the following.

Proposition 1. *Consider that Assumption 1 holds. Assume that the quadrotor is equipped with a set of sensors to measure y_i, $i = 1, \cdots, 6$. Assume that the quadrotor flies over a surface with enough visual characteristics and there is a region W_i containing the pixel location $x_p = y_p = 0$, where the optical flow is constant. Then, there exist constants α_1, α_2 and a matrix Γ such that the observer dynamic (29) complimentarily fuses the available measurements and the observer error $\tilde{V}^b$ exponentially converges to zero.*

Proof. The function $\gamma(\bar{y}_1, \bar{y}_4)$ in (24) performs the complementary fusion of the available measurements directly related to the quadrotor translational velocity. From this point, the observer design follows the lines of Theorem 1.

To analyze the observation error's stability properties, consider the following Lyapunov function [2]

$$\mathcal{V} = \frac{1}{2}(\tilde{V}^b)^\top \Gamma_1 \tilde{V}^b \tag{31}$$

with $\Gamma_1 \in \mathbb{R}^{3\times3}$ a diagonal positive definite matrix. Thus, one has

$$\lambda_m(\Gamma_1)\|\tilde{V}^b\|^2 \leq \mathcal{V} \leq \lambda_M(\Gamma_1)\|\tilde{V}^b\|^2 \tag{32}$$

with $\lambda_m(A)$ and $\lambda_M(A)$ the smallest and greatest eigenvalues of any matrix A.

The time derivative of (31) along the trajectories of the observer error dynamic (30) is given by

$$\dot{\mathcal{V}} = (\tilde{V}^b)^\top \Gamma_1 \left[-S(y_3)\tilde{V}^b + \frac{1}{\alpha_1 + \alpha_2}\Gamma(\alpha_1 \bar{H}_1 + \alpha_2 \bar{H}_2)\tilde{V}^b \right] \tag{33}$$

Since $S(y_3)$ is a skew symmetric matrix, it follows that

$$\dot{\mathcal{V}} = (\tilde{V}^b)^\top \left[\Gamma_1 \frac{1}{\alpha_1 + \alpha_2}\Gamma(\alpha_1 \bar{H}_1 + \alpha_2 \bar{H}_2) \right] \tilde{V}^b \tag{34}$$

It is straightforward to verify that there exist α_1, α_2, Γ, and Γ_1 such that the matrix

$$P = \left[\Gamma_1 \frac{1}{\alpha_1 + \alpha_2}\Gamma(\alpha_1 \bar{H}_1 + \alpha_2 \bar{H}_2) \right] \tag{35}$$

is negative definite. Thus,

$$\dot{\mathcal{V}} \leq -\lambda_m(P)\|\tilde{V}^b\|^2 \leq -\frac{\lambda_m(P)}{\lambda_M(\Gamma_1)}\mathcal{V} \tag{36}$$

and the proof is concluded. □

4. Results

This section presents a semi-realistic numerical simulation study to validate the theoretical developments of the previous section. First, the available measurements—the body axes acceleration and the optical flow—must be computed. Then, the proposed observer is implemented. It is essential to underscore that the Parrot Minidrone simulator provided by MATLAB-Simulink incorporates realistic quadrotor dynamics and sensor models. The works in [8–10] illustrate that the experimental implementation is straightforward after performing numerical simulations using this simulator; see also the work in [34], where semi-realistic simulations are performed.

The Parrot Minidrone's physical characteristics and the camera specifications are summarized in Table 1.

Table 1. Drone's physical characteristics and camera specifications.

Dimensions	(13.2 × 13.2 × 4.1) cm.
Weight	(63) g.
Motor type	Brushless (4).
Flight time	(8) min.
Camera	(Monocular / Vertical).
Shutter speed	(60) fps.
Resolution	(120 × 160) px.
Focal length	(1) mm.

4.1. Determination of the Parameter μ

Note that in Equation (24), two essential measurements to reconstruct the quadrotor velocity are $\bar{y}_1$ and $\bar{y}_4$. Table 1 shows that the quadrotor's mass is available, but the parameter μ is not. Hence, the quadrotor follows a circular trajectory, recording, along the $0X^b$ axis, its acceleration a_x^b and velocity u to determine μ from the following relationship

$$\mu = -\frac{a_x^b}{u}m \tag{37}$$

It is important to highlight that to compute μ, the measured acceleration a_x^b was filtered using a low-pass first-order filter, as recommended in [23]. Then, the value of μ was computed as the average value of the μ values obtained during the flight. Figure 4 shows (white line) the recorded acceleration a^b and (blue line) the reconstructed acceleration; this is,

$$\bar{a}_x^b = -\frac{\mu}{m}u \tag{38}$$

Hence, for this quadrotor, one has $\mu = 0.0035$. The parameter μ is related to the blade's aerodynamic profile and induced drag forces. This positive constant is known in helicopter literature as blade drag [23,35].

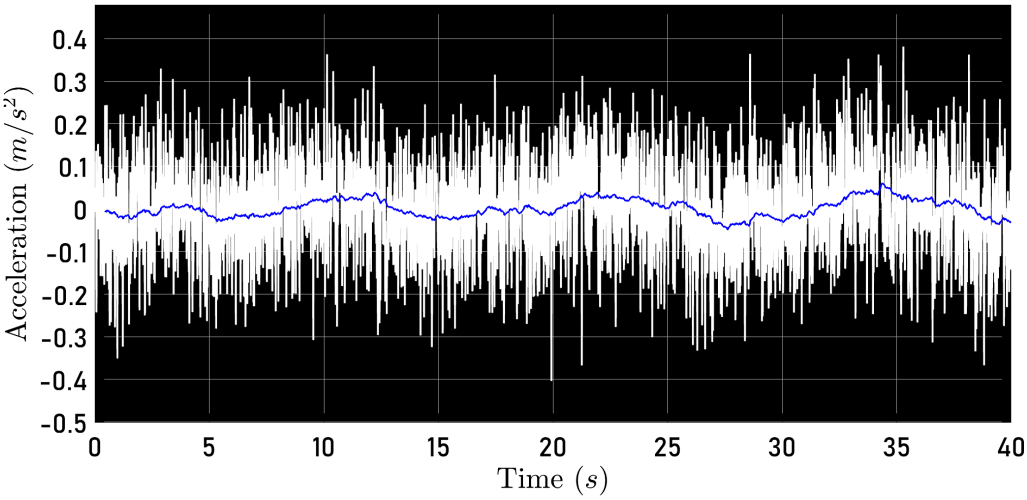

Figure 4. Measured acceleration a_x^b and reconstructed acceleration $\bar{a}_x^b$.

Note that this is an open-loop reconstruction so it is not expected that a_x^b and $\bar{a}_x^b$ are exactly coincident. However, as reported in [17,23], this procedure gives an adequate approximation of μ.

4.2. Optical Flow Algorithm Design

The optical flow estimation algorithm is implemented in the Parrot Minidrone Competition simulation environment, specifically in the Image Processing subsystem in the Flight Control System block. The optical flow estimation algorithm is performed as follows: (a) The monocular camera information that arrives in the Y1UY2V format is transformed into the RGB format. (b) The image in RGB format is transformed to Grayscale and filtered using an FIR (Finite Response Impulse) filter represented as a 2D coefficient matrix or a pair of separable filter coefficient vectors. (c) The filtered image is used to estimate the optical flow, employing the Lucas–Kanade method. The block implementing the Lucas–Kanade method delivers the pixel displacement per frame as a complex number for each image's pixel.

These calculated displacements are multiplied by the corresponding number of frames per second (intrinsic parameter of the camera) to obtain the pixel velocity. Finally, the pixel velocities are subjected to statistical processing consisting of selecting a region of interest on the image plane, which, considering Assumption 1, corresponds to the image origin. Figure 5 shows a block diagram of the optical flow algorithm.

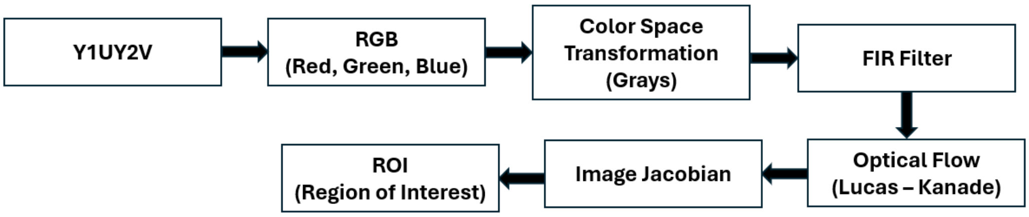

Figure 5. Optical flow block design.

4.3. Quadrotor Trajectories

To test the proposed observer, two trajectories are considered. In the first one, the quadrotor tracks a circle while in the second one, the quadrotor visits two waypoints. Figure 6 shows the quadrotor's path during the circle-tracking flight at a constant altitude. Figure 7 depicts the quadrotor's path during the two-way point flight, which is at a constant altitude once again.

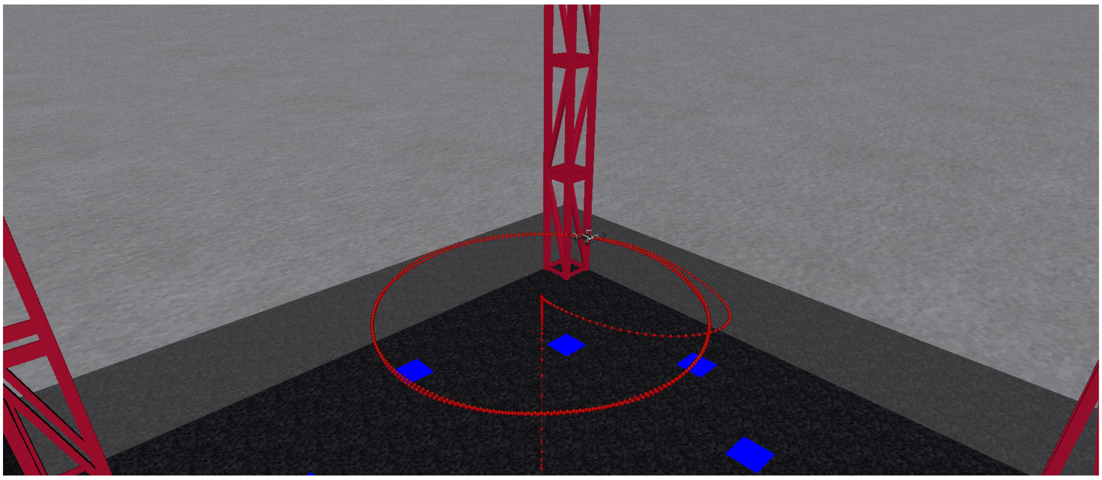

Figure 6. Aircraft tracking: circular-type trajectory.

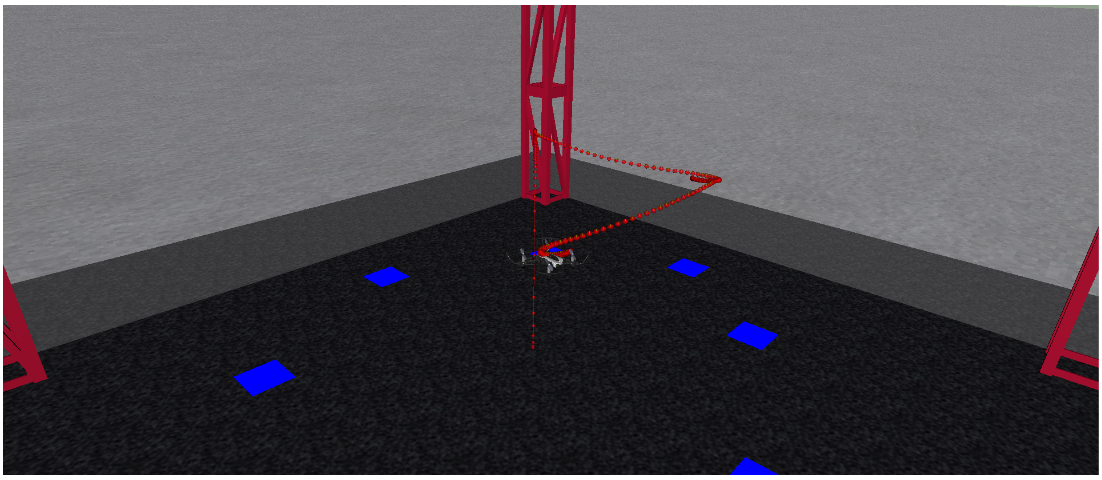

Figure 7. Aircraft tracking: square-type trajectory.

4.4. Measurements

Note that the parameter μ is required to implement the proposed observer to reconstruct the quadrotor-specific force. The quadrotor acceleration is obtained directly from the IMU implemented in the simulator. Figures 8 and 9 show the specific force along the $0X^b$ axis delivered by the IMU when the quadrotor tracks the circular and the square-type (the quadorotor moves first along the $0X^b$ axis and, 10 s after, moves along the $0Y^b$ axis) trajectories, respectively. It is essential to state that the IMU from Simulink is implemented considering that the accelerometers also measure the gravity force, contradicting Equation (4). Hence, the gravitational force is subtracted from the IMU's accelerometer measurement to match (4).

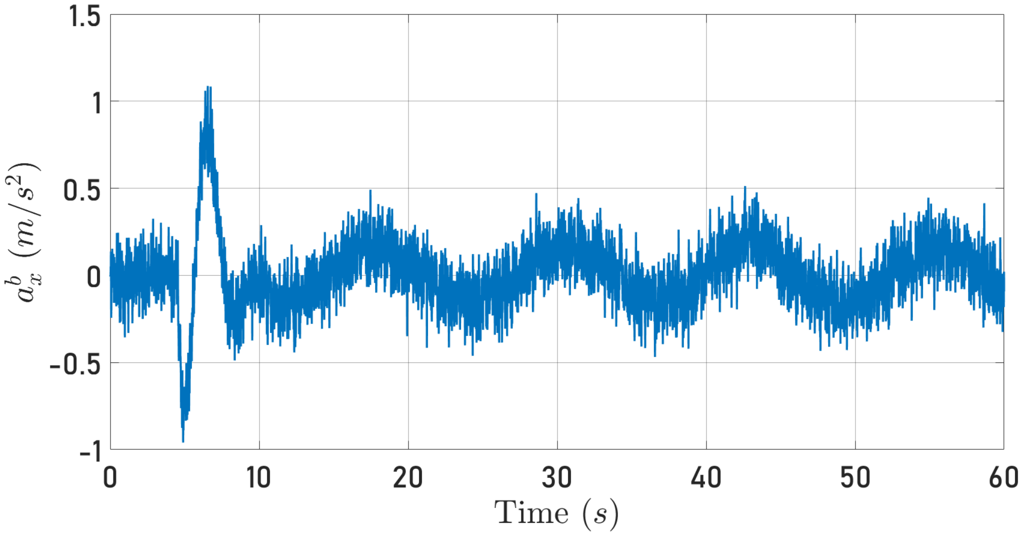

Figure 8. Specific force measured along the $0X^b$ axis while the quadrotor follows a circular trajectory.

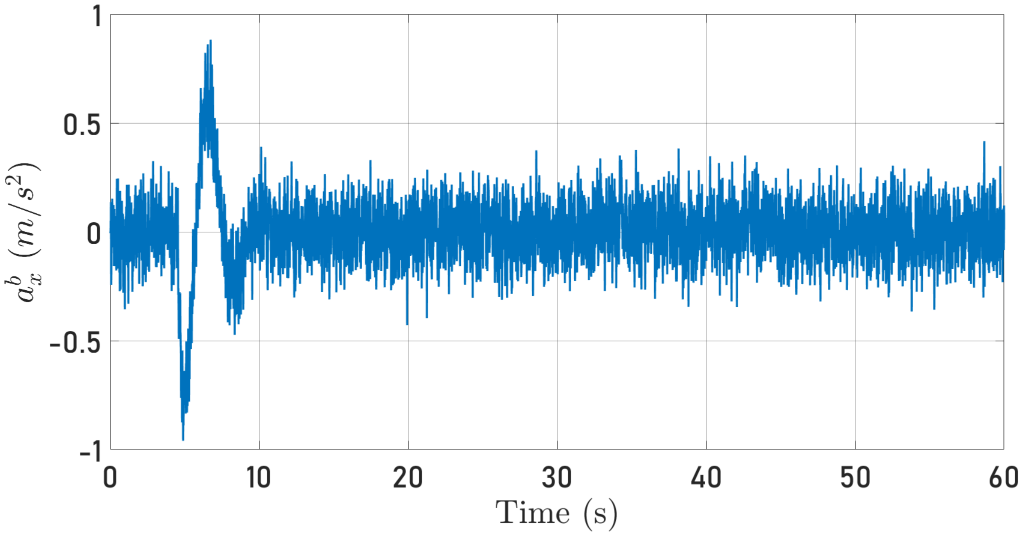

Figure 9. Specific force measured along the $0X^b$ axis while the quadrotor follows a square-type trajectory.

Figures 10 and 11 depict the quadrotor optical flow along the $0X^b$ axis computed as described in Section 4.2 when the quadrotor follows the circular and square-type trajectories, respectively.

From Equation (22), it is clear that the signals plotted in Figures 8–11 contain information about the translational velocity $\tilde{V}^b$. It is not evident to identify a relationship between the specific force and optical flow measurements. However, the proposed observer fuses the information and extracts the translational velocity $\tilde{V}^b$.

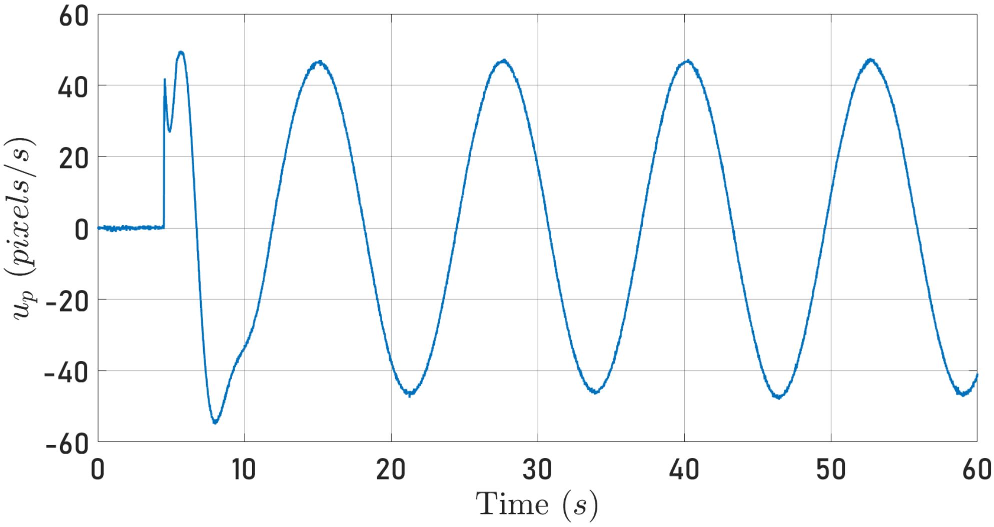

Figure 10. Computed optical flow along the $0X^b$ axis while the quadrotor tracks a circular trajectory.

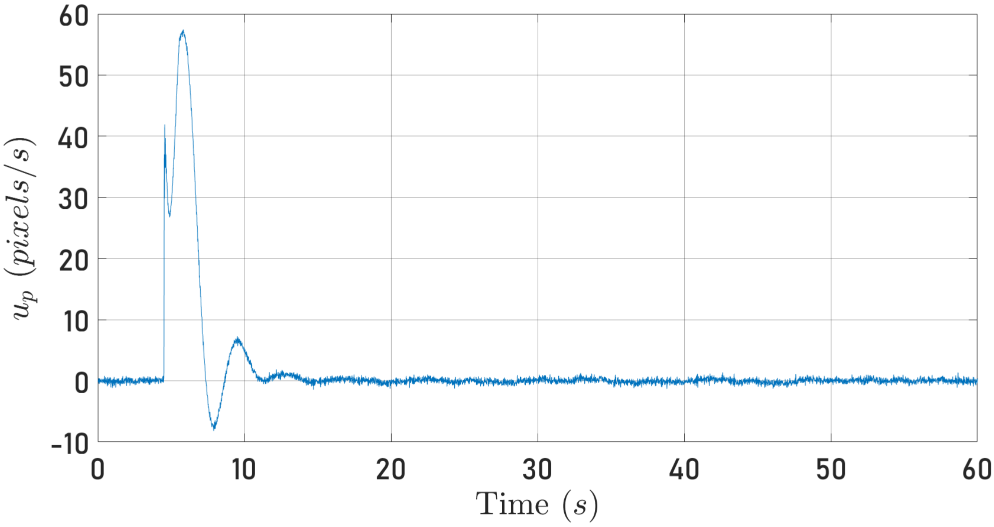

Figure 11. Computed optical flow along the $0X^b$ axis while the quadrotor tracks a square-type trajectory.

4.5. Observer Evaluation

The observer state dynamic described by Equation (29) is implemented inside the Parrot Mambo Minidrone simulator. The observed velocity

$$\hat{V}^b + \Gamma\sigma \tag{39}$$

is compared with the velocity computed by the internal algorithm of the Parrot Mambo Minidrone simulator, which employs a Kalman filter.

Figures 12 and 13 show the time history of the observed velocity along the $0X^b$ and $0Y^b$ axes, respectively, together with the speeds computed by the internal algorithm of the

Parrot Mambo Minidrone simulator when the quadrotor follows the circular trajectory. The observer gain matrices were fixed as follows

$$\Gamma = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad ; \quad \Gamma_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The scalar constants α_1 and α_2 were fixed at the following values

$$\alpha_1 = -13 \quad ; \quad \alpha_2 = 0.4 \tag{40}$$

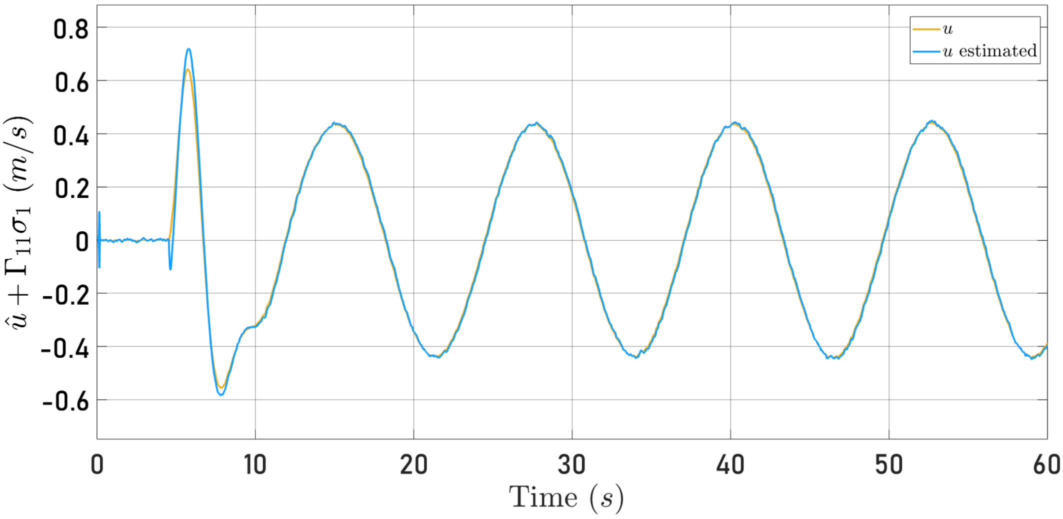

Figure 12. Observed speed $\hat{u} + \Gamma_{11}\sigma_1$ (blue line) and speed computed by the Parrot Mambo simulator algorithm u (yellow line).

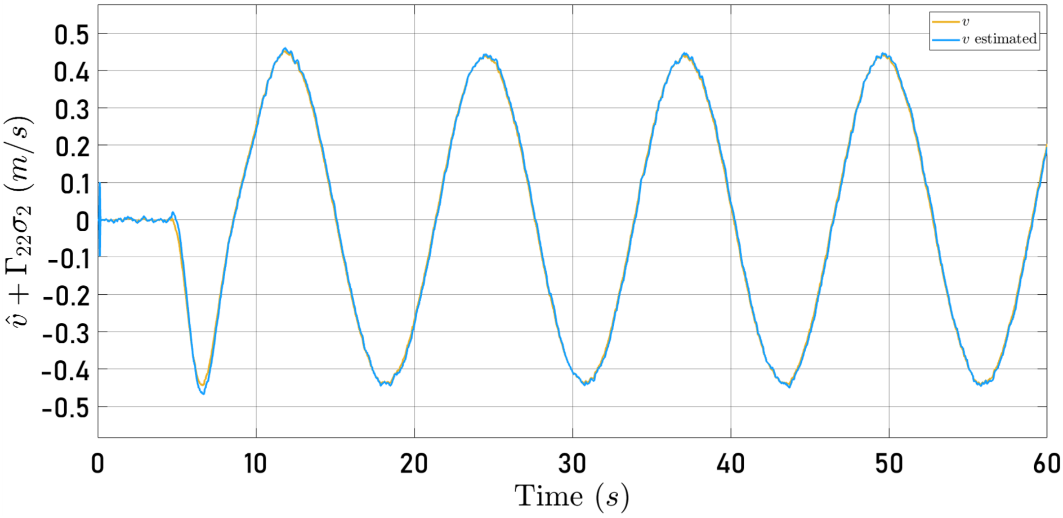

Figure 13. Observed speed $\hat{v} + \Gamma_{22}\sigma_2$ (blue line) and speed computed by the Parrot Mambo simulator algorithm v (yellow line).

Now, Figures 14 and 15 present the observer error behavior when the quadrotor follows the square-type trajectory. The speeds computed by the Parrot Mambo Minidrone algorithm are also shown.

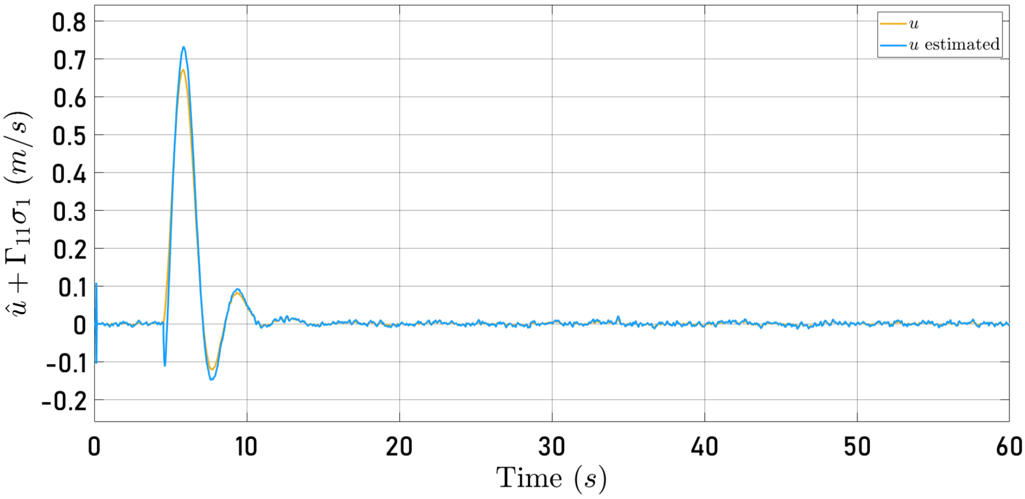

Figure 14. Observed speed $\hat{u} + \Gamma_{11}\sigma_1$ (blue line) and speed computed by the Parrot Mambo simulator algorithm u (yellow line).

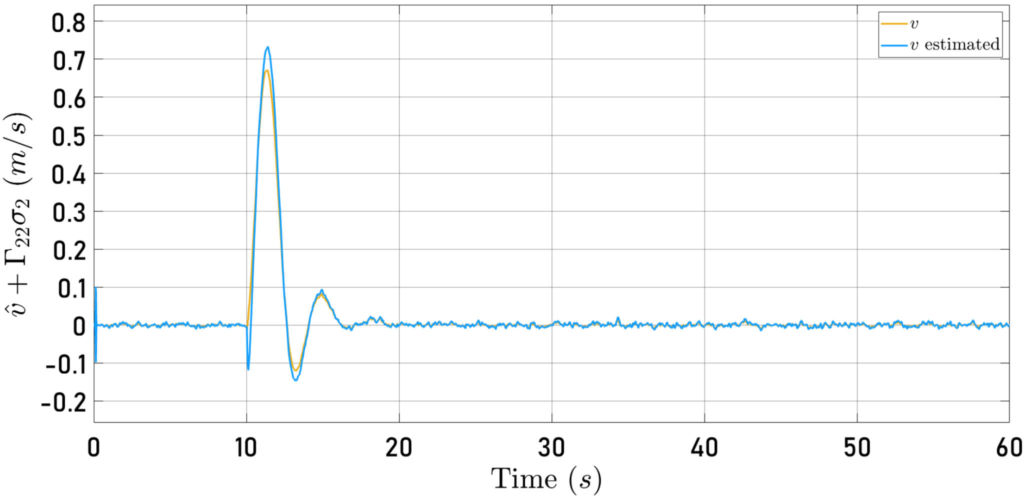

Figure 15. Observed speed $\hat{v} + \Gamma_{22}\sigma_2$ (blue line) and speed computed by the Parrot Mambo simulator algorithm v (yellow line).

Figures 12–15 illustrate that the selected observer gains performed adequately to identify both translational speeds.

Figures 16 and 17 present the observation errors to examine the proposed algorithm's performance more closely. Note that the observation errors only converge to a neighborhood of zero. This behavior results from the semi-realistic simulation that includes noise in the measurements. However, the strong result in Proposition 1 hints at this observer behavior.

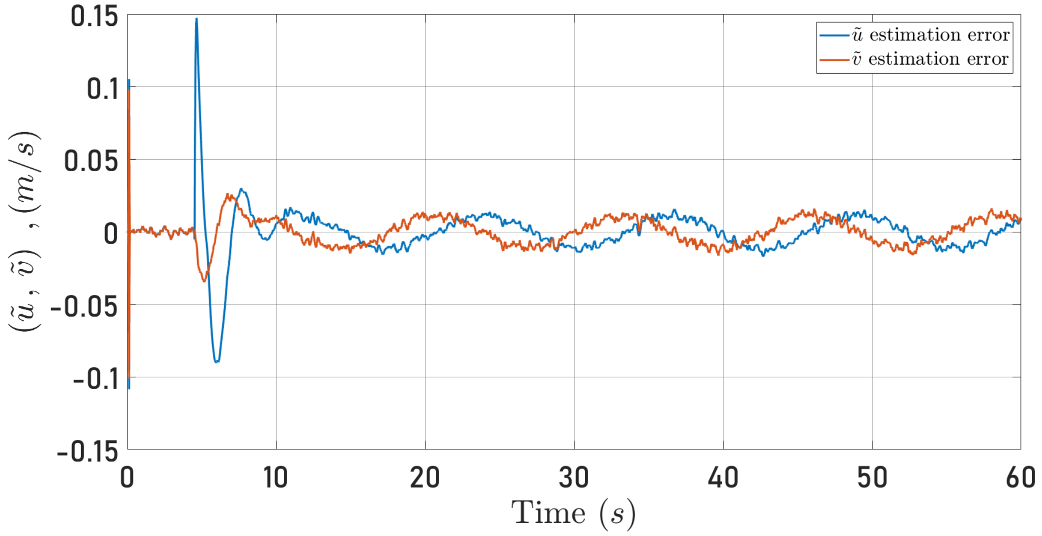

Figure 16. Observed speeds errors for the circular trajectory. $\tilde{u}$ (blue line), $\tilde{v}$ (red line).

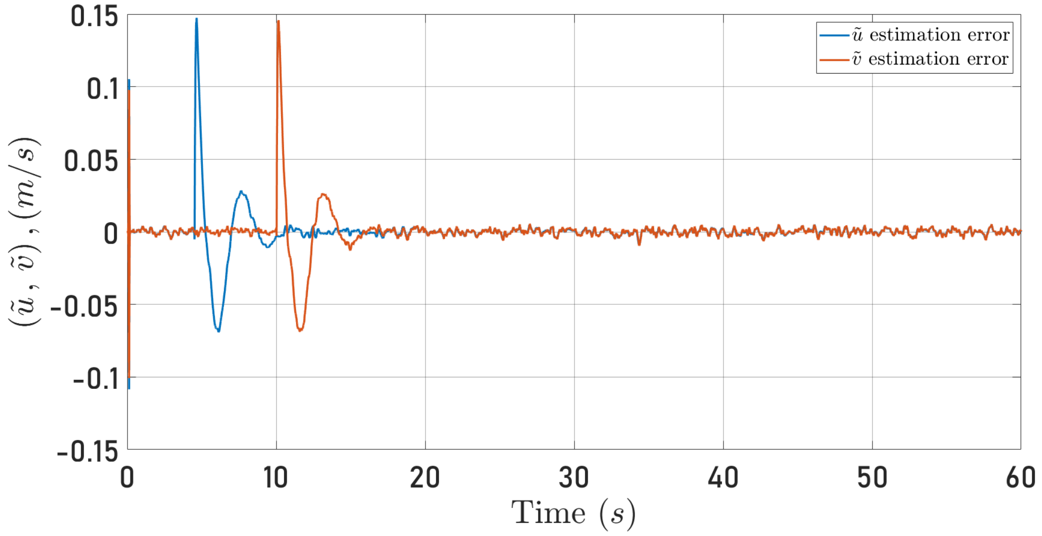

Figure 17. Observed speeds errors for the square-type trajectory. $\tilde{u}$ (blue line), $\tilde{v}$ (red line).

Remark 2. *Under mild assumptions, consider that measurement noise enters into the translational velocity dynamics* (16)*, as follows*

$$\dot{V}^b = gy_2^\top e_3 + y_1 - s(y_3)V^b + \delta(t) \tag{41}$$

with $\delta(t) \leq \bar{\delta}$ *a bounded time variant disturbance modeling the noise in all measurements. Then, the observer error dynamic becomes*

$$\dot{\tilde{V}}^b \quad = \quad -S(y_3)\tilde{V}^b + \frac{1}{\alpha_1+\alpha_2}\Gamma(\alpha_1\bar{H}_1 + \alpha_2\bar{H}_2)\tilde{V}^b - \delta(t) \tag{42}$$

Now, from the analysis in Proposition 1*, one obtains*

$$\begin{aligned} \dot{\mathcal{V}} &\leq -\lambda_m(P)\|\tilde{V}^b\|^2 + \lambda_M(\Gamma_1)\bar{\delta}\|\tilde{V}^b\| \\ &= -(1-\sigma)\lambda_m(P)\|\tilde{V}^b\|^2 - \sigma\lambda_m(P)\|\tilde{V}^b\|^2 + \lambda_M(\Gamma_1)\bar{\delta}\|\tilde{V}^b\| \end{aligned} \tag{43}$$

with $0 < \sigma < 1$*. Finally,*

$$\dot{\mathcal{V}} \quad \leq \quad -(1-\sigma)\|\lambda_m(P)\tilde{V}^b\|^2,\ \forall\,\|\tilde{V}^b\| \geq \tfrac{\lambda_M(\Gamma_1)\bar{\delta}}{\sigma\lambda_m(P)} \tag{44}$$

Hence, from Lemma 9.2 in [2], it follows that the observer error is ultimately bounded as observed in Figure 16. Note in (44), that the ultimate bound depends on the noise level modeled by $\bar{\delta}$*.*

A more challenging problem where the proposed observer design method may fail is the case where the noise enters the translational velocity dynamic as follows

$$\dot{V}^b = gy_2^\top e_3 + y_1 - s(y_3)V^b + \delta_1(t)V^b + \delta_2(t) \tag{45}$$

with $\delta_1(t) \leq \bar{\delta}_1$ *and* $\delta_2(t) \leq \bar{\delta}_2$ *bounded time variant disturbances.*

4.6. Observer Gains

A nonlinear time-varying model describes the observer error dynamics (see Equation (30)). Thus, determining adequate observer gains is a complex task; however, with some mild assumptions, at least locally, it is possible to determine a suitable observer gains combination. For example, assume that $y_3 \approx 0$; then, the observer error dynamics reduces to

$$\dot{\tilde{V}}^b \quad = \quad \begin{bmatrix} \frac{-\gamma_1(\alpha_1\frac{\mu}{m}+\alpha_2\frac{f}{\bar{y}_5})}{\alpha_1+\alpha_2} & 0 & 0 \\ 0 & \frac{-\gamma_2(\alpha_1\frac{\mu}{m}+\alpha_2\frac{f}{\bar{y}_5})}{\alpha_1+\alpha_2} & 0 \\ 0 & 0 & 2\gamma_3 \end{bmatrix} \tilde{V}^b \tag{46}$$

Hence, the following Eigenvalues locally shape the observer error dynamics

$$\begin{aligned} \lambda_1 &= \frac{-\gamma_1(\alpha_1\frac{\mu}{m}+\alpha_2\frac{f}{\bar{y}_5})}{\alpha_1+\alpha_2} \\ \lambda_2 &= \frac{-\gamma_2(\alpha_1\frac{\mu}{m}+\alpha_2\frac{f}{\bar{y}_5})}{\alpha_1+\alpha_2} \\ \lambda_3 &= 2\gamma_3 \end{aligned} \tag{47}$$

From Equation (47), the selection γ_3 follows trivially. Now, assuming that $\gamma_1 = \gamma_2$, the combination of γ_1 , α_1 and α_2 was selected as follows. For a fixed value of γ_1, the Eigenvalue $\lambda = \lambda_1$ was computed considering intervals for γ_1 and γ_2 that give a negative value, as illustrated in Figure 18. It is important to remember that a negative Eigenvalue will not guarantee an adequate observation of the quadrotor velocity due to numerical problems in the simulation.

In Figure 18, one can observe the optimal values region from the scalar constants, as well as the Eigenvalues, described previously, that were estimated with precision and accuracy. The displayed chromatic variety denotes the set of Eigenvalues λ, according to the different values that the scalar constants α_1 , α_2 can take. The last result Figure 19 shows the estimation errors analyzed only with respect to the x-axis because analogically, the y-axis presents the same behavior. As expected, increasing the value of λ decreases the observer error until the numerical errors appear at a specific value of λ, where the observer error diverges.

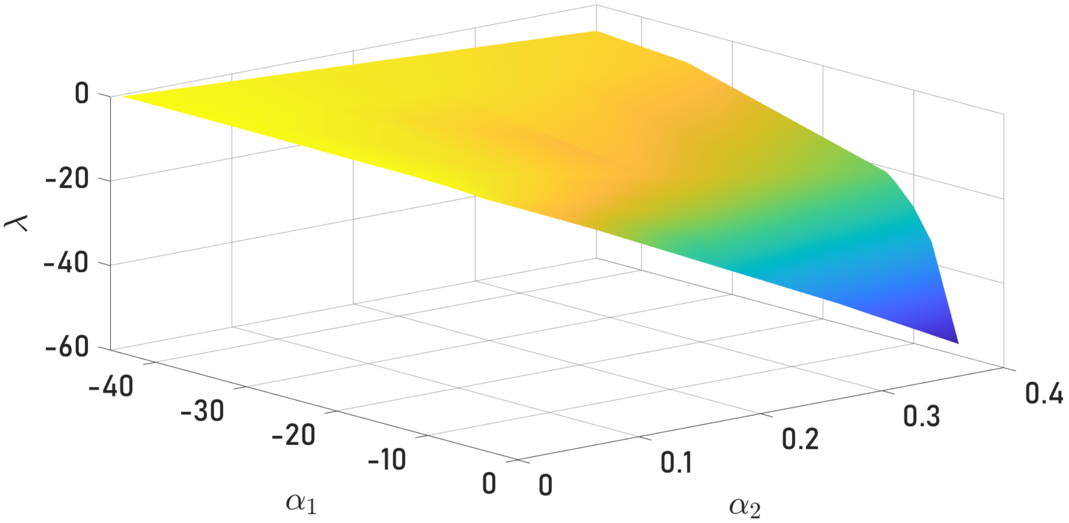

Figure 18. Eigenvalues for different combinations of gains α_1 and α_2.

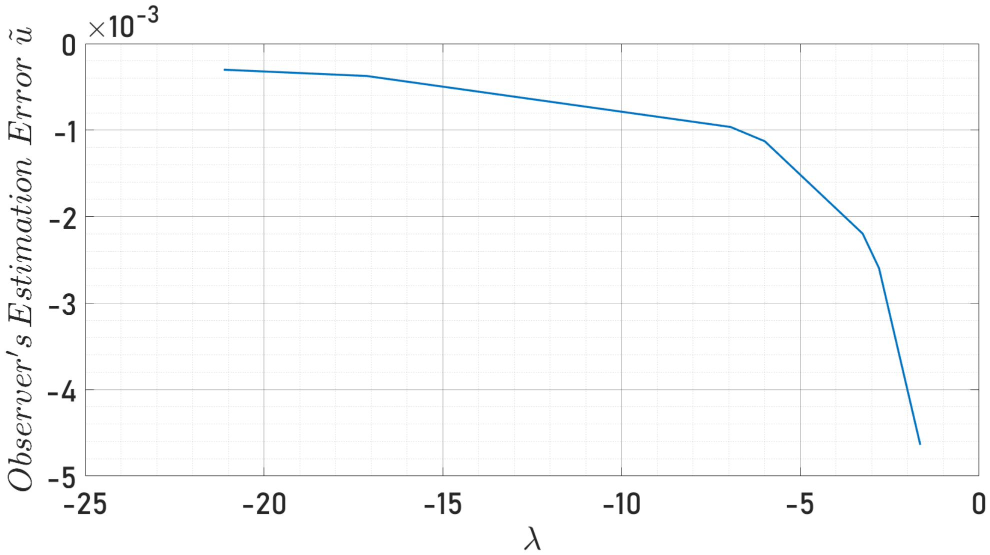

Figure 19. Eigenvalues vs. observer's estimation error in the $0X^b$ axis.

5. Conclusions

This article proposed a novel sensor fusion observation algorithm to observe the translational velocity using available measurements from onboard sensors for a quadrotor. Besides the successful implementation of the designed observation algorithm, the main contributions are listed next:

- The designed observer can estimate the linear speeds of the aircraft for different trajectories with precision.
- The optical flow was obtained without using image features or patterns compared to other research works.
- The application of Lyapunov's theory through a correct proposed Lyapunov function demonstrates asymptotic convergence to zero of the nonlinear observer error.
- As mentioned initially, this article was intended to compensate for the overall information loss in indoor flights by observing the vehicle's translational velocities. The observed velocity is precise enough so that the main objective has been successfully fulfilled.

- As previously evoked in the introduction, this research project surpassed the performance of the observer proposed in [17], mainly in flight trajectories, in which, at certain moments, the MAV does not make any displacements, remaining in stationary or hover flight, thanks to the incorporation of the *Optical Flow* into the observation algorithm.

Author Contributions: Conceptualization, H.R.-C.; methodology H.R.-C. and L.A.M.-P.; software J.R.M.-I.; validation H.R.-C., L.A.M.-P. and J.M.-U.; formal analysis H.R.-C.; investigation H.R.-C., L.A.M.-P., J.M.-U. and J.R.M.-I.; writing—original draft preparation J.R.M.-I. and J.M.-U.; writing—review and editing H.R.-C., L.A.M.-P., J.M.-U. and J.R.M.-I.; supervision H.R.-C. and L.A.M.-P. 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: Simulation files are available upon request to the first or second authors.

Acknowledgments: The first and second authors would like to acknowledge the finantial support from CONACyT.

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

References

1. Lyapunov, A.M. The general problem of the stability of motion. *Int. J. Control.* **1992**, *55*, 531–534. [CrossRef]
2. Khalil, H.K. *Control of Nonlinear Systems*; Prentice Hall: New York, NY, USA, 2002.
3. Astolfi, A.; Karagiannis, D.; Ortega, R. *Nonlinear and Adaptive Control with Applications*; Springer: Cham, Switzerland, 2008; Volume 187.
4. Van der Schaft, A. *L2-Gain and Passivity Techniques in Nonlinear Control*; Communications and Control Engineering; Springer: Cham, Switzerland, 2016.
5. Sasiadek, J.; Hartana, P. Sensor fusion for navigation of an autonomous unmanned aerial vehicle. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'04), New Orleans, LA, USA, 26 April–1 May 2004; Volume 4, pp. 4029–4034.
6. Jetto, L.; Longhi, S.; Venturini, G. Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots. *IEEE Trans. Robot. Autom.* **1999**, *15*, 219–229. [CrossRef]
7. Harriman, D.W.; Harrison, J.C. Gravity-induced errors in airborne inertial navigation. *J. Guid. Control. Dyn.* **1986**, *9*, 419–426. [CrossRef]
8. Noordin, A.; Mohd Basri, M.A.; Mohamed, Z. Adaptive PID Control via Sliding Mode for Position Tracking of Quadrotor MAV: Simulation and Real-Time Experiment Evaluation. *Aerospace* **2023**, *10*, 512. [CrossRef]
9. Okasha, M.; Kralev, J.; Islam, M. Design and Experimental Comparison of PID, LQR and MPC Stabilizing Controllers for Parrot Mambo Mini-Drone. *Aerospace* **2022**, *9*, 298. [CrossRef]
10. Rubio Scola, I.; Guijarro Reyes, G.A.; Garcia Carrillo, L.R.; Hespanha, J.P.; Burlion, L. A Robust Control Strategy With Perturbation Estimation for the Parrot Mambo Platform. *IEEE Trans. Control. Syst. Technol.* **2021**, *29*, 1389–1404. [CrossRef]
11. Zhu, C.; Chen, J.; Zhang, H. Attitude Control for Quadrotors Under Unknown Disturbances Using Triple-Step Method and Nonlinear Integral Sliding Mode. *IEEE Trans. Ind. Electron.* **2023**, *70*, 5004–5012. [CrossRef]
12. Naseer, F.; Ullah, G.; Siddiqui, M.A.; Jawad Khan, M.; Hong, K.S.; Naseer, N. Deep Learning-Based Unmanned Aerial Vehicle Control with Hand Gesture and Computer Vision. In Proceedings of the 2022 13th Asian Control Conference (ASCC), Jeju, Republic of Korea, 4–7 May 2022; pp. 1–6. [CrossRef]
13. Nascimento, T.; Saska, M. Embedded fast nonlinear model predictive control for micro aerial vehicles. *J. Intell. Robot. Syst.* **2021**, *103*, 1–11. [CrossRef]
14. McGuire, K.; de Croon, G.; de Wagter, C.; Remes, B.; Tuyls, K.; Kappen, H. Local histogram matching for efficient optical flow computation applied to velocity estimation on pocket drones. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 3255–3260. [CrossRef]
15. Hoang, M.L.; Carratù, M.; Paciello, V.; Pietrosanto, A. Fusion Filters between the No Motion No Integration Technique and Kalman Filter in Noise Optimization on a 6DoF Drone for Orientation Tracking. *Sensors* **2023**, *23*, 5603. [CrossRef]
16. Benzemrane, K.; Damm, G.; Santosuosso, G. Nonlinear adaptive observer for Unmanned Aerial Vehicle without GPS measurements. In Proceedings of the 2009 European Control Conference (ECC), Budapest, Hungary, 23–26 August 2009; pp. 597–602. [CrossRef]

17. Gómez-Casasola, A.; Rodríguez-Cortés, H. Scale Factor Estimation for Quadrotor Monocular-Vision Positioning Algorithms. *Sensors* **2022**, *22*, 8048. [CrossRef] [PubMed]
18. Borup, K.T.; Fossen, T.I.; Johansen, T.A. A nonlinear model-based wind velocity observer for unmanned aerial vehicles. *IFAC-PapersOnLine* **2016**, *49*, 276–283. [CrossRef]
19. Hosen, J.; Helgesen, H.H.; Fusini, L.; Fossen, T.I.; Johansen, T. A Vision-aided Nonlinear Observer for Fixed-wing UAV Navigation. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, San Diego, CA, USA, 4–8 January 2016; pp. 1–19. [CrossRef]
20. He, Y.; Wang, D.; Huang, F.; Zhang, R.; Min, L. Aerial-Ground Integrated Vehicular Networks: A UAV-Vehicle Collaboration Perspective. *IEEE Trans. Intell. Transp. Syst.* **2023**. [CrossRef]
21. Kalenberg, K.; Müller, H.; Polonelli, T.; Schiaffino, A.; Niculescu, V.; Cioflan, C.; Magno, M.; Benini, L. Stargate: Multimodal Sensor Fusion for Autonomous Navigation On Miniaturized UAVs. *IEEE Internet Things J.* **2024**. [CrossRef]
22. Mahony, R.; Kumar, V.; Corke, P. Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. *IEEE Robot. Autom. Mag.* **2012**, *19*, 20–32. [CrossRef]
23. 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]
24. Ma, Y.; Soatto, S.; Košecká, J.; Sastry, S. *An Invitation to 3-D Vision: From Images to Geometric Models*; Springer: Cham, Switzerland, 2004; Volume 26.
25. 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. [CrossRef]
26. Weiss, L.; Sanderson, A.; Neuman, C. Dynamic sensor-based control of robots with visual feedback. *IEEE J. Robot. Autom.* **1987**, *3*, 404–417. [CrossRef]
27. Wu, Y. *Optical Flow and Motion Analysis*; Advanced Computer Vision Notes Series 6; 2001. Available online: http://www.eecs.northwestern.edu/~yingwu/teaching/EECS432/Notes/optical_flow.pdf (accessed on 1 May 2024).
28. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. In Proceedings of the IJCAI'81: 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–28 August 1981; Volume 2, pp. 674–679.
29. Barron, J.L.; Fleet, D.J.; Beauchemin, S.S. Performance of optical flow techniques. *Int. J. Comput. Vis.* **1994**, *12*, 43–77. [CrossRef]
30. Spagl, M. Optische Positions-und Lagebestimmung einer Drohne im Geschlossenen Raum. Master's Thesis, Hochschule Rosenheim Unoiversity of Applied Sciences, Rosenheim, Germany, 2021.
31. Mitchell, H.B. *Multi-Sensor Data Fusion: An Introduction*; Springer Science & Business Media: Berlin, Germany, 2007.
32. Durrant-Whyte, H.F. Sensor models and multisensor integration. *Int. J. Robot. Res.* **1988**, *7*, 97–113. [CrossRef]
33. Boudjemaa, R.; Forbes, A. *Parameter Estimation Methods in Data Fusion*; NPL Report CMSC 38/04; 2004. Available online: https://eprintspublications.npl.co.uk/2891/1/CMSC38.pdf (accessed on 1 May 2024).
34. Noordin, A.; Mohd Basri, M.A.; Mohamed, Z. Position and attitude tracking of MAV quadrotor using SMC-based adaptive PID controller. *Drones* **2022**, *6*, 263. [CrossRef]
35. Bramwell, A.R.S.; Balmford, D.; Done, G. *Bramwell's Helicopter Dynamics*; Elsevier: Amsterdam, The Netherlands, 2001.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.