sensors

MDPI

Article

A Drag Model-LIDAR-IMU Fault-Tolerance Fusion Method for Quadrotors

Pin Lyu, Bingqing Wang, Jizhou Lai *, Shichao Liu and Zhimin Li

Navigation Research Center, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; lvpin@nuaa.edu.cn (P.L.); wangbingqing@nuaa.edu.cn (B.W.); liushichao@nuaa.edu.cn (S.L.); lizhimin@nuaa.edu.cn (Z.L.)

* Correspondence: laijz@nuaa.edu.cn

Received: 5 September 2019; Accepted: 1 October 2019; Published: 8 October 2019

check for updates

Abstract: In this paper, a drag model-aided fault-tolerant state estimation method is presented for quadrotors. Firstly, the drag model accuracy was improved by modeling an angular rate related item and an angular acceleration related item, which are related with flight maneuver. Then the drag model, light detection and ranging (LIDAR), and inertial measurement unit (IMU) were fused based on the Federal Kalman filter frame. In the filter, the LIDAR estimation fault was detected and isolated, and the disturbance to the drag model was estimated and compensated. Some experiments were carried out, showing that the velocity and position estimation were improved compared with the traditional LIDAR/IMU fusion scheme.

Keywords: model-aided navigation; fault-tolerant estimation; improved drag model; quadrotor

1. Introduction

In recent years, quadrotors have been widely used in many fields, such as agriculture, industry, and ecology [1]. Navigation systems calculate attitude, velocity, and position, which are important for stable control. For quadrotors, inertial navigation systems (INS) and global positioning system (GPS) integration systems are mostly used. However, GPS cannot be used in an indoor environment because of signal interference. In a GPS denied environment, vision [2] or light detection and ranging (LIDAR) [3] based navigation methods are usually adopted.

Vision navigation is based on computer vision algorithms. The environment is captured by the camera, then the relative motion between the camera and the environment is estimated. However, the estimation accuracy is usually affected by light interference [4]. In some dark and enclosed environments, such as caves and tunnels [5], a LIDAR-based navigation method is usually adopted. LIDAR is an active sensor and can measure ranges between itself and the environment. The motion state is estimated by using the simultaneous localization and mapping (SLAM) algorithm [6].

LIDAR can be divided into 2 dimensional (2D) LIDAR [7] and 3 dimensional (3D) LIDAR [8]. The 2D type can measure a two-dimensional plane, and the 3D type can measure three-dimensional space. Due to size and weight limitations, 2D LIDAR is usually adopted by quadrotors. Because quadrotors fly in a 3D environment, the classical 2D LIDAR SLAM may lead to estimation error [9]. It is usually assumed that the environment is formed by collections of vertical walls [10]. In some complex or sparse-feature environments, the state estimation accuracy will decrease. As a result, the robustness of the SLAM algorithm in these environments is now a challenging issue [11].

Dynamic model-aided navigation is a novel and developed method. The dynamic model of plants, which describes the relationship between the plant motion, control input, and surrounding environment, is fused with navigation sensors, improving the navigation accuracy and reliability. The dynamic model-aided navigation method has been used for aircraft [12], vehicles [13], and underwater robots [14]. The fusion schemes for different plants are different because they should be designed based on the model characteristics.

As for the fixed wing aircraft, the whole dynamic model (including the thrust, drag, and moments) is considered [15]. The dynamic model is fused with the INS. Experiment results show that the scheme is useful for the low-cost micro-electromechanical systems (MEMS) INS and can improve the navigation accuracy [16]. As the state estimation using the dynamic model is a dead reckoning process, the estimation accuracy is affected by the model parameter uncertainty [17].

The quadrotor is an underactuated system. The horizontal motion is coupled with the attitude motion, which is usually estimated by fusing gyros and accelerometers [18], and the drag is proportional to the velocity [19]. In recent years, drag model-aided navigation has been widely studied. [20] proposed a drag model/INS fusion scheme. The velocity and attitude estimation are improved compared with the pure INS scheme. [21] showed that the velocity estimation accuracy can be kept the same while the frequency of position correction decreases. A unified model technique was used by [22]. The state prediction of the dynamic model and INS are fused, and the procedure of the traditional Kalman filter is simplified. A dynamic model/INS/GPS fusion scheme was proposed by [23]. When GPS is available, the INS/GPS fusion results are used to identify the model parameters. When GPS is denied, the dynamic model is fused with INS. The dynamic model/INS/vision sensor fusion scheme is adopted by the AR. Drone [24]. The accelerometer's bias can be estimated online. In [25], the dynamic model was used to estimate the scale factor of monocular vision. In [26], a dynamic model/optical flow/inertial sensor fault-tolerant fusion method was proposed. The faults of the above sensors can be detected. The navigation accuracy can be retained in the case of failure of one sensor.

In this paper, a drag model-LIDAR-IMU fusion scheme (Figure 1) is proposed for quadrotors in an indoor environment. The contributions of the paper can be summarized as:

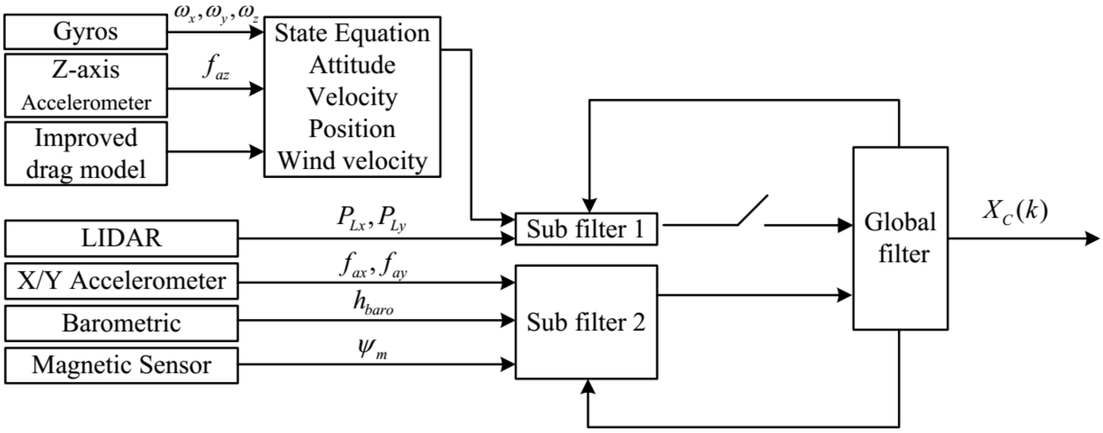


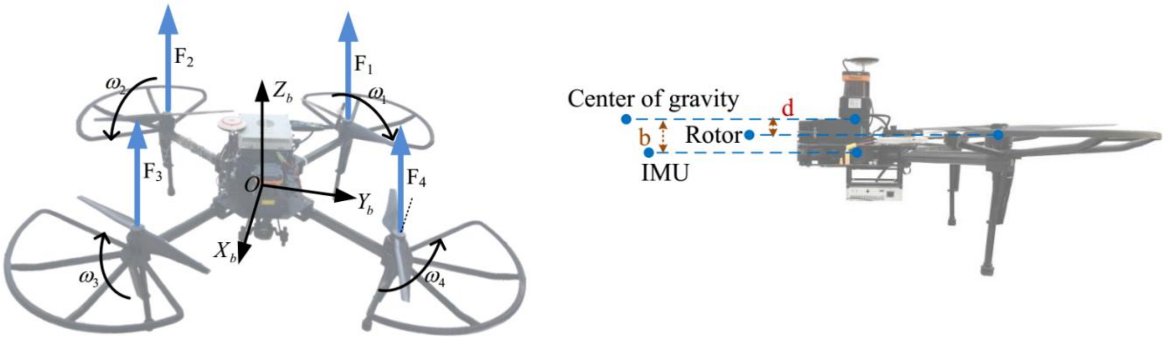
Figure 1. The architecture of the proposed fault-tolerant filter.

(1) An improved drag model of quadrotors is proposed. The traditional drag model only contains the velocity related item. Its accuracy is affected by maneuver flight [27]. In this paper, some attitude motion related items are considered to improve the model accuracy.

(2) A fault-tolerant state estimation is realized for quadrotors. The failures of LIDAR and drag model are both considered. If the LIDAR SLAM accuracy decreases due to environmental disturbance, it is detected and isolated from the global filter. The drag model accuracy may be affected by the wind. In the filter, the wind velocity is treated as a state and can be estimated online. Therefore, the accuracy decrease brought by the wind is suppressed.

2. Improved Drag Model of Quadrotor

In this paper, the body coordinate system (b-frame) is defined as the front-right-down frame (Figure 2), and the navigation coordinate system (n-frame) is defined as the local north-east-down frame. In the dynamic model-aided navigation method, the model accuracy affects the navigation performance. Therefore, the drag model will be studied in this section.



(**a**) Oblique view of the quadrotor (**b**) Side view of half the quadrotor

Figure 2. Quadrotor structure diagram.

2.1. Drag Modeling of Quadrotor

The traditional drag model can be expressed as [18]

$$\begin{cases} D_x = k_x(V^b_{nbx} - V^b_{wx}) \\ D_y = k_y(V^b_{nby} - V^b_{wy}) \end{cases} \tag{1}$$

where D_x and D_y represent the x-axis and y-axis drag force resolved in the b-frame. V^b_{nbx} and V^b_{nby} are the components of $\boldsymbol{V^b_{nb}}$, which is the linear velocity of the b-frame with respect to the n-frame resolved in the b-frame. V^b_{wx} and V^b_{wy} are the components of $\boldsymbol{V^b_w}$, which is the wind velocity resolved in the b-frame. k_x and k_y are the drag coefficients.

Equation (1) means that the drag is proportional to the airspeed. If the wind is ignored, Equation (1) is transformed to

$$\begin{cases} D_x = k_x V^b_{nbx} \\ D_y = k_y V^b_{nby} \end{cases}, \tag{2}$$

Because the drag force can be estimated by the accelerometer, it can be derived as

$$\begin{cases} f^b_{nbx} m = k_x V^b_{nbx} \\ f^b_{nby} m = k_y V^b_{nby} \end{cases}, \tag{3}$$

where f^b_{nbx} and f^b_{nby} are the x-axis and y-axis accelerations resolved in the b-frame, which can be obtained from the outputs of the accelerometers. It can be seen that V^b_{nbx} and V^b_{nby} are proportional to f^b_{nbx} and f^b_{nby}.

In the traditional INS algorithm, velocity is calculated by integrating the accelerometers' outputs. Using the drag model, the velocity can be directly estimated from the accelerometers. Therefore, the velocity error can be bounded.

In this paper, two additional factors are taken into account. Firstly, the traditional drag force model is derived based on the characteristics of one blade, and the quadrotor is treated as a mass point. However, there exists distances among the rotors, accelerometers, and center of gravity (CoG), shown in Figure 2. d represents the distance between the rotor and the CoG, and b represents the distance between the accelerometers and the CoG. When the quadrotor pitches or rolls, a tangential acceleration between the accelerometer and the center of gravity is introduced. Therefore, Equation (3) transforms to

$$\begin{cases} \left(f_{nbx}^{b}-\dot{\omega}_{nby}^{b}/b-f_{x0}\right)m=k_{x}V_{nbx}^{b} \\ \left(f_{nby}^{b}-\dot{\omega}_{nbx}^{b}/b-f_{y0}\right)m=k_{y}V_{nby}^{b} \end{cases}, \tag{4}$$

where $\boldsymbol{\omega}_{nb}^{b}=\begin{bmatrix} \omega_{nbx}^{b} & \omega_{nby}^{b} & \omega_{nbz}^{b} \end{bmatrix}$ is the angular rate of the b-frame with respect to the n-frame resolved in the b-frame, which can be estimated from gyros' outputs. f_{x0} and f_{y0} are the biases of the x- and y-axis accelerometers.

Secondly, in the traditional drag model, the blade flapping effect is ignored, which is an important aspect of quadrotor dynamics [28]. The flapping effect causes the blade rotation plane to tilt, and the blade velocity can be expressed as [29]

$$\mathbf{V}_{ri}=\mathbf{V}_{nb}^{b}+\boldsymbol{\omega}_{nb}^{b}\times\mathbf{L}_{i}, \tag{5}$$

where $\mathbf{V}_{ri}$ is the ith rotor's velocity, and $\mathbf{L}_{i}$ is the displacement between the rotor and the CoG. Therefore, the drag force of the *i*th rotor can be expressed as

$$\mathbf{D}_{ri}=k\mathbf{V}_{ri}=k\left(\mathbf{V}_{nb}^{b}+\boldsymbol{\omega}_{nb}^{b}\times\mathbf{L}_{i}\right), \tag{6}$$

Then the drag force of the whole quadrotor can be derived as

$$\begin{aligned} \mathbf{D} &= \sum_{i=1}^{4}\mathbf{D}_{ri} \\ &= \sum_{i=1}^{4}k\mathbf{V}_{ri} \\ &= k\left(4\mathbf{V}_{nb}^{b}+\sum_{i=1}^{4}\boldsymbol{\omega}_{nb}^{b}\times\mathbf{L}_{i}\right) \\ &= 4k\left(\mathbf{V}_{nb}^{b}+\begin{bmatrix} d\omega_{nby}^{b} & -d\omega_{nbx}^{b} & 0 \end{bmatrix}^{T}\right) \end{aligned} \tag{7}$$

Combining Equations (4) and (7), the improved drag model can be expressed as

$$\begin{cases} \left(f_{nbx}^{b}-\dot{\omega}_{nby}^{b}/b\right)m=k'_{x}\left(V_{nbx}^{b}+d\omega_{nby}^{b}\right) \\ \left(f_{nby}^{b}-\dot{\omega}_{nbx}^{b}/b\right)m=k'_{y}\left(V_{nby}^{b}-d\omega_{nbx}^{b}\right) \end{cases}, \tag{8}$$

Then the velocity can be derived as

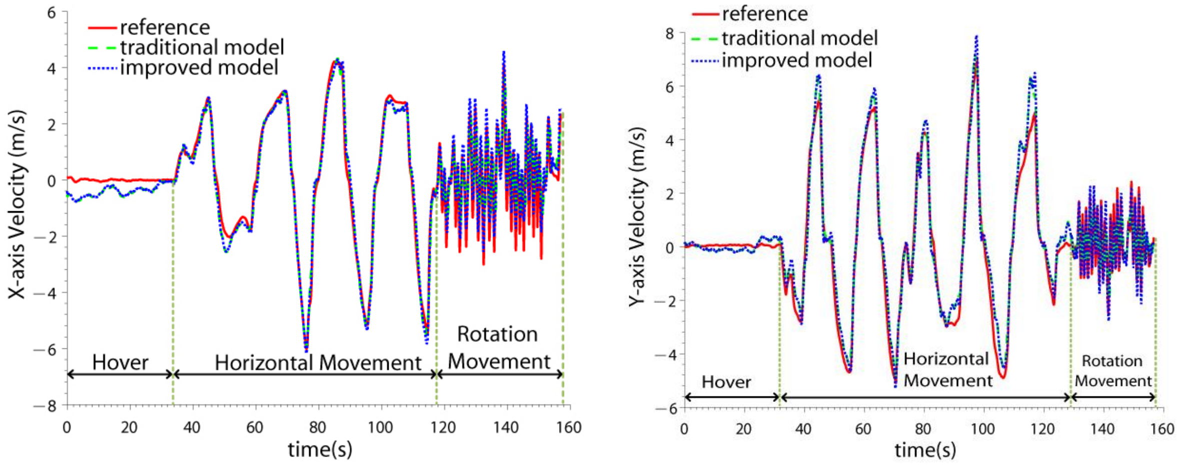
$$\begin{cases} V_{nbx}^{b}=k_{x0}+k_{x1}f_{nbx}^{b}+k_{x2}\dot{\omega}_{nby}^{b}+k_{x3}\omega_{nby}^{b} \\ V_{nby}^{b}=k_{y0}+k_{y1}f_{nby}^{b}+k_{y2}\dot{\omega}_{nbx}^{b}+k_{y3}\omega_{nbx}^{b} \end{cases}, \tag{9}$$

where k_{x0}, k_{x1}, k_{x2}, k_{x3}, k_{y0}, k_{y1}, k_{y2}, and k_{y3} are constant coefficients.

It can be seen that there are four items in the improved drag model: the accelerometer bias, the velocity related item, the angular rate related item, and the angular acceleration related item.

2.2. Test of Drag Model Accuracy

Some experiments were done to verify the superiority of the improved model. The quadrotor was controlled to do different maneuvers, including hover, horizontal flight, and attitude rotation. Five experiments were conducted in a no wind environment. The least square method was used to identify the model parameters. The velocities estimated by the traditional model and the improved model were compared. The velocity obtained by GPS was treated as a reference. In the improved model, the angular acceleration was obtained by differentiating the gyros' signals. Because the difference operation introduces noise to the signal, it was smoothed before being used. The accelerometers' outputs were also used as inputs of the models. The experimental results are shown in Figure 3 and Table 1. Figure 3 is the velocity estimation result of one experiment, and Table 1 is the statistical results of the five experiments.



(**a**) The x-axis velocity estimation results (**b**) The y-axis velocity estimation results

Figure 3. The comparison between the velocities estimated by different drag models.

Table 1. The velocity RMSE (root mean square error) comparison of different drag models.

State	**X-axis Velocity RMSE (m/s)**		**Y-axis Velocity RMSE (m/s)**	
	Traditional Drag Model	**Improved Drag Model**	**Traditional Drag Model**	**Improved Drag Model**
Hover	0.455	0.443	0.190	0.189
Horizontal movement	0.288	0.267	0.573	0.554
Rotation movement	0.908	0.655	0.837	0.534

From the experiments, it can be seen that during the hover or horizontal flight, the accuracy of the two models are almost the same. When the quadrotor does an attitude motion, the accuracy of the improved model is better than the traditional model by 1.38 times in x-axis and 1.56 times in y-axis.

3. Drag Model-LIDAR-IMU Fusion Scheme

In this section, the fault-tolerant fusion scheme is studied. The drag model, IMU, and LIDAR were fused through a Federal Kalman filter (FKF). The filter can deal with the disturbances to the LIDAR and drag model.

3.1. Quadrotor Dynamic Equation

According to the drag model (considering the wind effect) and the INS algorithm, the velocity differential equation can be expressed as

$$\begin{bmatrix} \dot{V}^b_{nbx} \\ \dot{V}^b_{nby} \\ \dot{V}^b_{nbz} \end{bmatrix} = \begin{bmatrix} V^b_{nbx} \\ V^b_{nby} \\ V^b_{nbz} \end{bmatrix} \times \begin{bmatrix} \omega^b_{nbx} \\ \omega^b_{nby} \\ \omega^b_{nbz} \end{bmatrix} + \begin{bmatrix} (V^b_{nbx} - V^b_{wx} - k_{x0} - k_{x2}\dot{\omega}^b_{nby} - k_{x3}\omega^b_{nby})/k_{x1} \\ (V^b_{nby} - V^b_{wy} - k_{y0} - k_{y2}\dot{\omega}^b_{nbx} - k_{y3}\omega^b_{nbx})/k_{y1} \\ f^b_{nbz} \end{bmatrix} + C^b_n \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \quad (10)$$

where f^b_{nbz} is the z-axis acceleration resolved in the b-frame, C^b_n is the coordinate transformation matrix from the n-frame to the b-frame, and g is the gravitational acceleration.

The attitude is described through quaternion, and the attitude differential equation can be expressed as

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = 0.5 \begin{bmatrix} 0 & -\omega_{nbx}^b & -\omega_{nby}^b & -\omega_{nbz}^b \\ \omega_{nbx}^b & 0 & \omega_{nbz}^b & -\omega_{nby}^b \\ \omega_{nby}^b & -\omega_{nbz}^b & 0 & \omega_{nbx}^b \\ \omega_{nbz}^b & \omega_{nby}^b & -\omega_{nbx}^b & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}, \tag{11}$$

where q_0, q_1, q_2, q_3 stand for the quaternion.

The position differential equation is

$$\begin{bmatrix} \dot{P}_N \\ \dot{P}_E \\ \dot{P}_D \end{bmatrix} = C_b^n \begin{bmatrix} V_{nbx}^b \\ V_{nby}^b \\ V_{nbz}^b \end{bmatrix}, \tag{12}$$

The wind velocity is regarded as constant velocity expressed as

$$\begin{bmatrix} \dot{V}_{wx}^b \\ \dot{V}_{wy}^b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{13}$$

3.2. Fault-Tolerant Filter Design

The architecture of the filter is shown in Figure 1. It contains a main filter and two sub-filters, which are denoted as C1 and C2, respectively. The wind velocity affects the drag model accuracy, thus it is included in the state vector, which is chosen as

$$\mathbf{x_c} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 & V_{nbx}^b & V_{nby}^b & V_{nbz}^b & P_N & P_E & P_D & V_{wx}^b & V_{wy}^b \end{bmatrix}^T, \tag{14}$$

The input vector is defined as

$$\mathbf{u} = \begin{bmatrix} \omega_x & \omega_y & \omega_z & f_{az} & \dot{\omega}_x & \dot{\omega}_y & g \end{bmatrix}^T, \tag{15}$$

where ω_x, ω_y, and ω_z are the x, y, and z axis gyros' outputs, f_{az} is the z-axis accelerometer's output, and g is the gravitational acceleration.

The state equation can be derived from (10)–(13) and expressed as

$$\dot{\mathbf{x}}{=}f(\mathbf{x},\mathbf{u}){+}\mathbf{Gw}, \tag{16}$$

where $\mathbf{G}$ is the noise transition matrix, and $\mathbf{w}$ is the state noise matrix.

The measurements of the two sub-filters are chosen as

$$\begin{cases} \mathbf{z}_1 = \begin{bmatrix} P_{Lx} & P_{Ly} \end{bmatrix} \\ \mathbf{z}_2 = \begin{bmatrix} f_{ax} & f_{ay} & \psi_m & h_{baro} \end{bmatrix} \end{cases}, \tag{17}$$

where P_{Lx} and P_{Ly} are the position estimated by LIDAR SLAM method, f_{ax} and f_{ay} are the outputs of the x- and y-axis accelerometers, ψ_m is the yaw estimated by the magnetic sensor, and h_{baro} is the height estimated by the barometer.

The measurement equations of the two sub-filters are expressed as

$$\begin{cases} \mathbf{z}_1 = h_{c1}(\mathbf{x_c}) + \mathbf{V_{C1}} \\ \mathbf{z}_2 = h_{c2}(\mathbf{x_c}) + \mathbf{V_{C2}} \end{cases}, \tag{18}$$

where $\mathbf{V}_{\mathbf{C}1}$ and $\mathbf{V}_{\mathbf{C}2}$ are the measurement noise matrices of C1 and C2. The updating progress of the FKF can be referred to in [30] and is not described here.

3.3. Fault Detection of LIDAR SLAM

In this paper, the LIDAR SLAM fault is considered. The LIDAR SLAM accuracy is affected by the feature salience of the environment. In some sparse-feature environments, the accuracy is low. In that case, the LIDAR SLAM result should be cut off from the global filter. Although the wind introduces disturbance to the drag model, it is estimated in the filter, so the drag model fault is not considered.

The chi-square test [31] is used for the fault detection of sub-filter C1. The statistics parameter is defined as

$$\lambda_{C1}(k)\mathbf{=}\mathbf{r}_{\mathbf{C}1}(k)\mathbf{P}_{\mathbf{C}1\mathbf{r}}(k)^{-1}\mathbf{r}_{\mathbf{C}1}(k)^{T}, \tag{19}$$

where $\mathbf{r}_{\mathbf{C}1}(k)$ is the residual of C1, and $\mathbf{P}_{\mathbf{C}1\mathbf{r}}(k)$ is the covariance matrix of $\mathbf{r}_{\mathbf{C}1}(k)$. They are defined as

$$\mathbf{r}_{\mathbf{C}1}(k) = \mathbf{z}_1(k) - h_{c1}(\mathbf{x}_\mathbf{c}(k|k-1)), \tag{20}$$

$$\mathbf{P}_{\mathbf{C}1\mathbf{r}}(k) = \mathbf{H}_{\mathbf{C}1}(k)\mathbf{P}_{\mathbf{C}1}(k|k-1)\mathbf{H}_{\mathbf{C}1}(k)^{T} + \mathbf{R}_{\mathbf{C}1}(k), \tag{21}$$

where $\mathbf{x}_\mathbf{c}(k|k-1)$ is the prediction state, $\mathbf{P}_{\mathbf{C}1}(k|k-1)$ is the covariance matrix of $\mathbf{x}_\mathbf{c}(k|k-1)$, $\mathbf{H}_{\mathbf{C}1}(k)$ is Jacobian matrix and calculated by $\mathbf{H}_{\mathbf{C}1}(k) = \frac{\partial h_{c1}(\mathbf{x}_\mathbf{C}(k|k-1))}{\partial \mathbf{x}_\mathbf{C}(k|k-1)}$, and $\mathbf{R}_{\mathbf{C}1}(k)$ is the measuring noise variance matrix.

When there is no fault, $\lambda_{C1}(k)$ follows a chi-square distribution [30]. The fault detection function can be constructed as

$$T_D(k) = \begin{cases} 1 & \lambda_{C1}(k) > \tau_D \\ 0 & \lambda_{C1}(k) < \tau_D \end{cases}, \tag{22}$$

where τ_D is the threshold. $T_D(k) = 1$ means that the LIDAR SLAM error increases and should be isolated from the filter.

3.4. Observability Analysis

The observability is an important indicator for a filter. It can reflect whether the states can be estimated. In this paper, the lie deviation method [32] is adopted for observability analysis. Two cases are considered: (1) no fault occurs; (2) the position supplied by LIDAR SLAM is faulty.

When there is no fault, the rank of the observability matrix is 12. It means that all the states are observable. However, when the position measurement is not available the rank is 8. It means that 4 states are not observable. Using the null space analysis method [33], the unobservable states are $\begin{bmatrix} P_N & P_E & V_{wx}^b & V_{wy}^b \end{bmatrix}$, which are the horizontal position and wind velocity.

4. Experiments and Analysis

In this section, experiments designed and carried out to test the proposed method are described. The following cases were considered:

(1) The navigation result in the LIDAR SLAM failure case. The navigation performance of the proposed method needs to be tested.
(2) The test when the quadrotor does an attitude maneuver. The proposed drag model improves compared with the traditional model, so the navigation accuracy should be tested.
(3) The test under wind. The wind introduces interference to the model, so the navigation accuracy in windy environment should be considered.

4.1. Test Setup

The test platform was built based on a DJI M100 quadrotor. The quadrotor specification is shown in Table 2. The navigation system included an IMU, a magnetic sensor, a barometer, and a 2D

LIDAR. The navigation result was outputted to the autopilot N1 and served as the control algorithm. The update rates of IMU, magnetic sensor, and barometer were 50 Hz, and the update rate of the LIDAR SLAM was 10 Hz.

Table 2. Unmanned aerial vehicle (UAV) technical features.

Technical Features	Description
Airframe	DJI-M100 Arm length 0.65 m
Autopilot	DJI N1
2D LIDAR	Hokuyou TM-30LX, Scanning range 30 m
Navigation processor	DJI Manifold

The test scheme is shown in Figure 4. The experiments were carried out in an underground garage. A total station (Leica MS60) was used as the position reference. The position accuracy was better than 0.01 m.

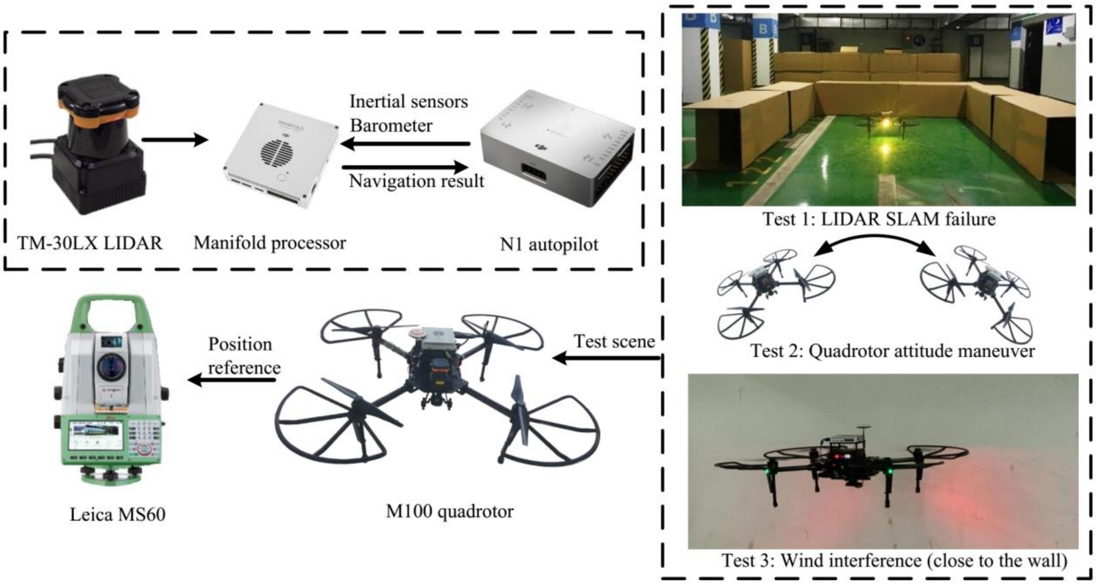
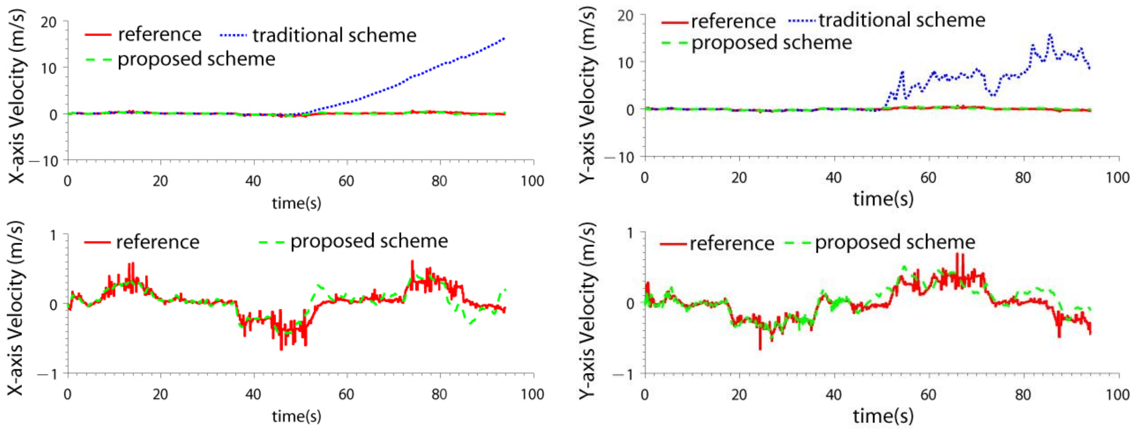


Figure 4. The test scheme.

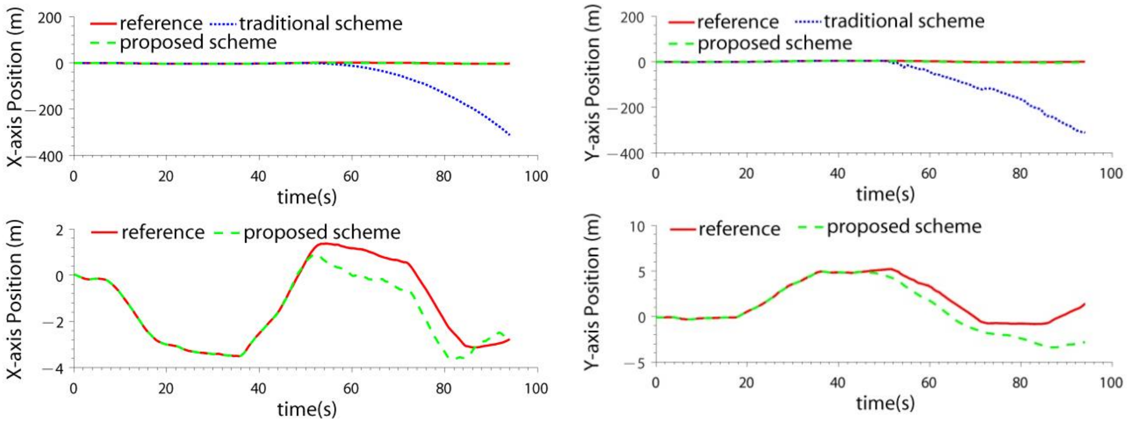
4.2. Test in LIDAR SLAM Failure Case

In this paper, the ICP (Iterated Closest Points) LIDAR SLAM algorithm [34,35] was adopted. It is a classical SLAM method. Because the SLAM method for 2D LIDAR is based on the 2.5D assumption [9,10], when there exists a step change in the environment, position estimation error may be introduced. In the test, some carton boxes were placed to construct the environment changes (shown in Figure 4). When the quadrotor flew across the boxes, the LIDAR SLAM failed. The velocity and position estimation results of two schemes are compared: the IMU/LIDAR fusion scheme and the proposed drag model-LIDAR-IMU fusion scheme (shown in Figures 5 and 6). The fault detection result is shown in Figure 7. The RMSE of velocity and position is shown in Table 3.



(**a**) The x-axis estimation result (**b**) The y-axis estimation result

Figure 5. The velocity estimation result in light detection and ranging (LIDAR) simultaneous localization and mapping (SLAM) failure case.



(**a**) The x-axis estimation result (**b**) The y-axis estimation result

Figure 6. The position estimation result in LIDAR SLAM failure case.

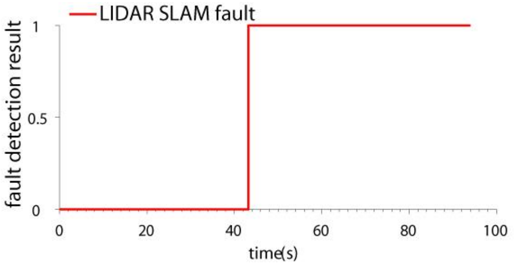


Figure 7. The fault detection results in LIDAR SLAM failure case.

Table 3. The RMSE (root mean square error) comparison of different schemes.

State	X-axis Velocity RMSE (m/s)	Y-axis Velocity RMSE (m/s)	X-axis Position RMSE (m)	Y-axis Position RMSE (m)
Traditional Scheme	8.020	7.503	121.832	147.137
Proposed Scheme	0.147	0.147	0.899	1.885

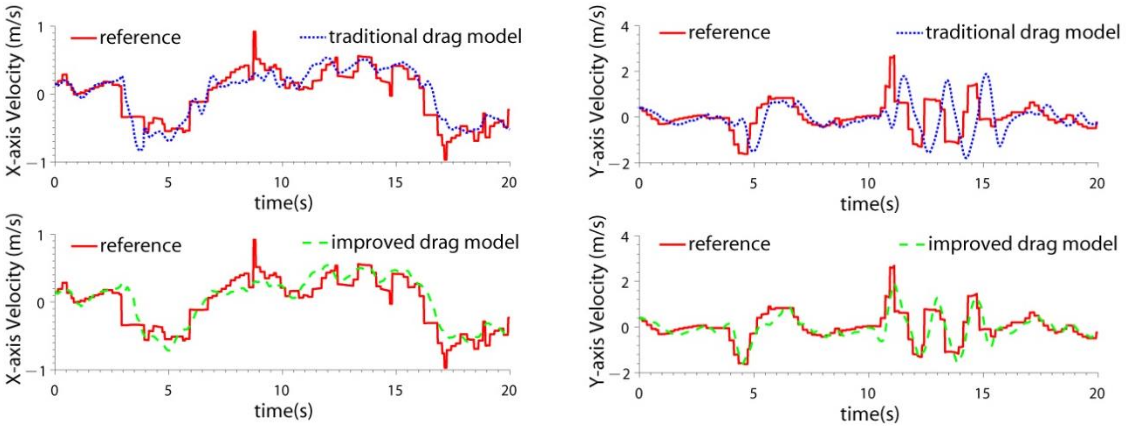
From the experimental results, it can be seen that:

(1) When the quadrotor flew over the boxes, the LIDAR SLAM algorithm failed due to a step environment change. The LIDAR SLAM failure can be detected and isolated by both the two schemes.

(2) When the LIDAR was isolated from the filter, the IMU/LIDAR fusion scheme degraded to the pure INS scheme. The navigation accuracy improved by introducing the drag model. The velocity error was bounded, and the positioning error also significantly decreased. The x-axis and y-axis velocity accuracies improved by 54.6 times and 51.0 times, respectively. The x-axis and y-axis position accuracies improved by 135.5 times and 78.1 times, respectively.

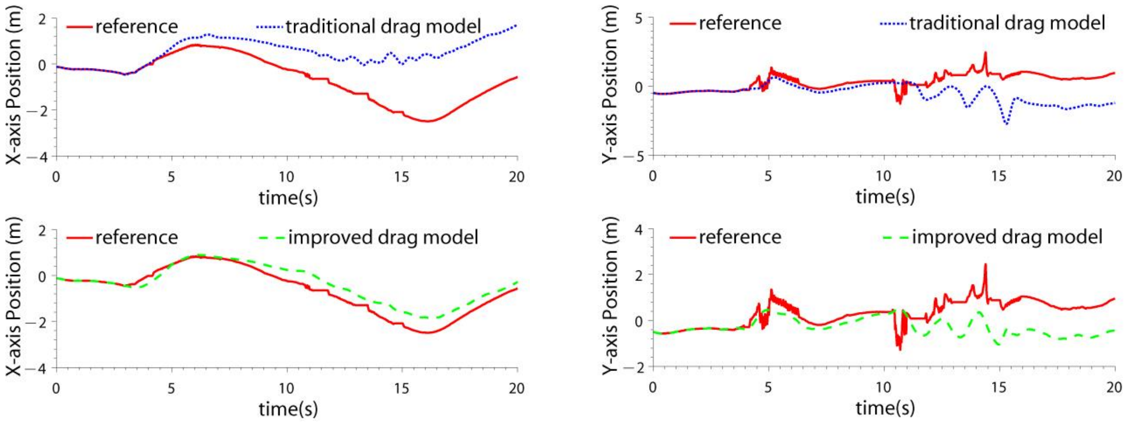
4.3. Quadrotor Attitude Maneuver Test

The quadrotor was controlled to do attitude maneuvers. The traditional drag model and the improved drag model were compared. The navigation results are shown in Figure 8, Figure 9, and Table 4. The fault detection result is shown in Figure 10.



(**a**) The x-axis estimation result (**b**) The y-axis estimation result

Figure 8. The velocity estimation result in quadrotor attitude maneuver case.



(**a**) The x-axis estimation result (**b**) The y-axis estimation result

Figure 9. The position estimation result in quadrotor attitude maneuver case.

Table 4. The RMSE (root mean square error) comparison of different drag model.

State	X-axis Velocity RMSE (m/s)	Y-axis Velocity RMSE (m/s)	X-axis Position RMSE (m)	Y-axis Position RMSE (m)
Traditional Drag Model	0.192	0.975	1.631	1.388
Improved Drag Model	0.188	0.422	0.414	0.952

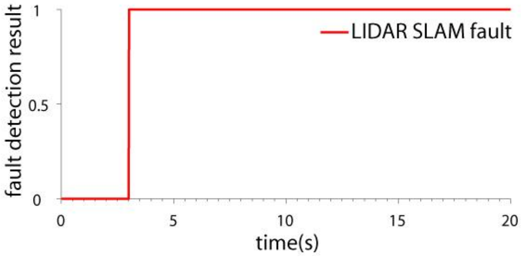


Figure 10. The fault detection results in quadrotor attitude maneuver case.

From the experimental results, it can be seen that:

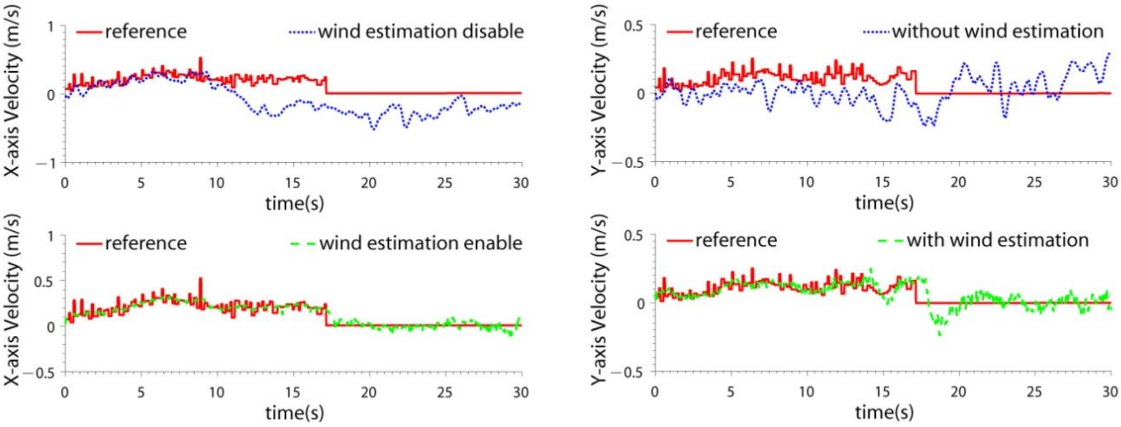
(1) When the quadrotor completed attitude maneuvers, the LIDAR SLAM accuracy decreased and failed. This was due to the mismatch of the LIDAR scanned points.

(2) In the test, the quadrotor completed an attitude maneuver in the y-axis, so the y-axis velocity accuracy improved by 2.3 times using the improved model, while the x-axis velocity accuracies of the two models were almost the same. The percentage increase of the x-axis position accuracy (3.9 times) was larger than the y-axis position (1.5 times), that is because the velocity errors of the y-axis velocity were offset after the integration.

(3) It was noticed that the accuracy improvement (2.3 times) was different from the test result of the y-axis velocity in Section 2.2, which was 1.56 times. That is because the flight maneuvers of the two tests were different, which affected the improvement degree.

4.4. Wind Interference Test

Although the experiments were carried out indoors, wind interference may still be present. For example, when the quadrotor flew close to the wall (Figure 4), the wind force generated by the blades reacted on the quadrotor. Because wind does not affect the LIDAR SLAM, the accuracy of LIDAR SLAM was not tested. The navigation results of two schemes were compared: the drag model-LIDAR-IMU fusion filter with and without wind estimation. The velocity estimation results are shown in Figure 11 and the RMSE is shown in Table 5. The wind estimation results are shown in Figure 12.



(**a**) The x-axis estimation result (**b**) The y-axis estimation result

Figure 11. The velocity estimation result in the wind interference case.

Table 5. The RMSE (root mean square error) comparison of different wind estimation.

State	X-axis Velocity RMSE (m/s)	Y-axis Velocity RMSE (m/s)
Wind Estimation Disable	0.264	0.141
Wind Estimation Enable	0.049	0.058

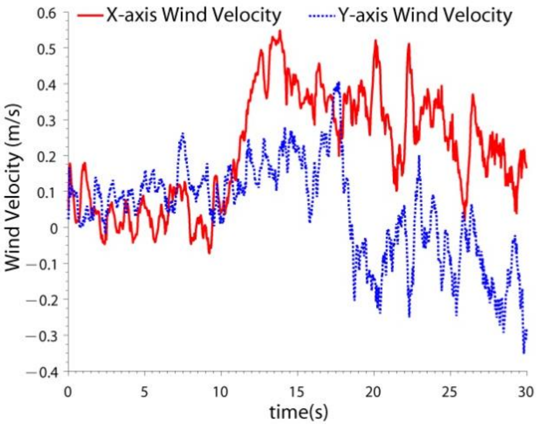


Figure 12. The wind estimation results.

From Figures 11 and 12 and Table 5, it can be seen that:

(1) When the quadrotor flew near the wall, the velocity estimation accuracy decreased. That is because the wind introduces interference to the drag model. If the wind velocity is included in the state, the wind can be estimated, and the interference can be partly compensated. The x-axis velocity accuracy improved by 5.4 times and the y-axis velocity accuracy improved by 2.4 times.

(2) It can be seen that when the quadrotor was away from the wall, the estimated wind velocity was small (0 s~10 s). When the quadrotor flew close to the wall, the wind became greater. Because wind is generated by the reaction of the rotating blades, the estimated wind is not constant.

5. Conclusions

A drag model-LIDAR-IMU fault-tolerant fusion method was proposed. An angular rate related item and an angular acceleration related item were added to the traditional drag model. The model accuracy during the maneuver flight improved. An FKF based fusion scheme was designed. The LIDAR SLAM estimation fault was detected and isolated from the filter, avoiding the disturbance to the navigation result. Compared with the traditional method, the velocity and position accuracy is improved by introducing the drag model. The wind velocity is included in the states and can be estimated online, and the filter is robust to wind interference.

Author Contributions: Funding acquisition, P.L. and J.L.; Investigation, P.L. and S.L.; Methodology, B.W. and J.L.; Software, S.L. and Z.L.; Supervision, J.L.; Validation, S.L. and Z.L.; Writing—review & editing, P.L. and J.L.

Funding: This work is partially supported by National Natural Science Foundation of China (61703207, 61973160), Natural Science Foundation of Jiangsu Province (BK20170801) and Aeronautical Science Foundation of China (2017ZC52017, 2018ZC52037).

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

References

1. Lyu, P.; Malang, Y.; Liu, H.; Lai, J.; Liu, J.; Jiang, B.; Qu, M.; Stephen, A.; Daniel, D.; Wang, Y. Autonomous cyanobacterial harmful algal blooms monitoring using multirotor UAS. *I. J. Remote Sens.* **2017**, *38*, 2818–2843. [CrossRef]
2. Almeshal, A.R.; Alenezi, M. A Vision-Based Neural Network Controller for the Autonomous Landing of a Quadrotor on Moving Targets. *Robotics* **2018**, *7*, 71. [CrossRef]
3. Wang, S.; Kobayashi, Y.; Ravankar, A.; Ravankar, A.; Emaru, T. A Novel Approach for Lidar-Based Robot Localization in a Scale-Drifted Map Constructed Using Monocular SLAM. *Sensors* **2019**, *19*, 2230. [CrossRef] [PubMed]
4. Yuan, C.; Lai, J.; Lyu, P.; Shi, P.; Zhao, W.; Huang, K. A Novel Fault-Tolerant Navigation and Positioning Method with Stereo-Camera/Micro Electro Mechanical Systems Inertial Measurement Unit (MEMS-IMU) in Hostile Environment. *Micromachines* **2018**, *9*, 626. [CrossRef] [PubMed]

5. Özaslan, T.; Loianno, G.; Keller, J.; Taylor, C.; kumar, V.; Vozencraft, J.; Hood, T. Autonomous Navigation and Mapping for Inspection of Penstocks and Tunnels With MAVs. *IEEE Robotics Autom. Lett.* **2017**, *2*, 1740–1747. [CrossRef]
6. Deschaud, J. IMLS-SLAM: Scan-to-Model Matching Based on 3D Data. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2480–2485.
7. Anderson, S.; MacTavish, K.; Barfoot, T. Relative continuous-time SLAM. *Int. J. Rob. Res.* **2015**, *34*, 1453–1479. [CrossRef]
8. Droeschel, A.; Nieuwenhuisen, M.; Beul, M.; Holz, D.; Stückler, J.; Behnke, S. Multilayered mapping and navigation for autonomous microaerial vehicles. *J. Field Rob.* **2016**, *33*, 451–475. [CrossRef]
9. Shen, S.; Michael, N.; Kumar, V. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 20–25.
10. Shen, S.; Mulgaonkar, Y.; Michael, N.; Kumar, V. Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 4974–4981.
11. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Rob.* **2016**, *32*, 1309–1332. [CrossRef]
12. Mohammadkarimi, H.; Nobahari, H. A Model Aided Inertial Navigation System for Automatic Landing of Unmanned Aerial Vehicles. *J. Navig.* **2018**, *65*, 183–204. [CrossRef]
13. Koppanyi, Z.; Navrátil, V.; Xu, H.; Toth, C.; Grejner-Brzezinska, D. Using Adaptive Motion Constraints to Support UWB/IMU Based Navigation. *J. Navig.* **2018**, *65*, 247–261. [CrossRef]
14. Karmoozdy, A.; Hashemi, M.; Salarieh, H. Design and practical implementation of kinematic constraints in Inertial Navigation System-Doppler Velocity Log (INS-DVL)-based navigation. *J. Navig.* **2018**, *65*, 629–642. [CrossRef]
15. Koifman, M.; Bar-Itzhack, I. Inertial navigation system aided by aircraft dynamics. *IEEE Trans. Control Syst. Technol.* **1999**, *4*, 487–793. [CrossRef]
16. Görcke, L.; Dambeck, G.; Holzapfel, F. Results of Model-Aided Navigation with Real Flight Data. In Proceedings of the 2014 International Technical Meeting of The Institute of Navigation, San Diego, CA, USA, 27–29 January 2014; pp. 407–412.
17. Khaghani, M.; Skloud, J. Autonomous Navigation of Small UAVs Based on Vehicle Dynamic Model. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS), Prague, The Czech Republic, 11–19 July 2016; pp. 117–122.
18. Crassidis, J.; Markley, F.; Cheng, Y. Survey of Nonlinear Attitude Estimation Methods. *J. Guidance Control Dyn.* **2007**, *30*, 12–28. [CrossRef]
19. Martin, P.; Salaün, E. The true role of accelerometer feedback in quadrotor control. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–7 May 2010; pp. 1623–1629.
20. Leishman, R.; Macdonald, J.; Beard, R.; McLain, T. Quadrotors and Accelerometers: State Estimation with an Improved Dynamic Model. *IEEE Control Syst. Mag.* **2014**, *34*, 28–41.
21. Macdonald, J.; Leishman, R.; Beard, R.; McLain, T. Analysis of an Improved IMU-Based Observer for Multirotor Helicopters. *J. Intell. Rob. Syst.* **2014**, *74*, 1049–1061. [CrossRef]
22. Crocoll, P.; Seibold, J.; Scholz, G.; Trommer, G. Model-Aided Navigation for a Quadrotor Helicopter: A Novel Navigation System and First Experimental Results. *J. Navig.* **2014**, *61*, 253–271. [CrossRef]
23. Zahran, S.; Moussa, A.; El-Sheimy, N.; Abu, B.S. Hybrid Machine Learning VDM for UAVs in GNSS-denied Environment. *J. Navig.* **2018**, *65*, 477–492. [CrossRef]
24. Bristeau, P.; Callou, F.; Vissière, D.; Petit, N. The Navigation and Control Technology inside the AR. Drone Micro UAV. *IFAC Proc. Volumes* **2011**, *44*, 1477–1484. [CrossRef]
25. Abeywardena, D.; Wang, Z.; Kodagoda, S.; Dissanayake, G. Visual-inertial fusion for quadrotor Micro Air Vehicles with improved scale observability. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 3148–3153.

26. Lyu, P.; Lai, J.; Liu, H.; Liu, J.; Chen, W. A Model-aided Optical Flow/Inertial Sensor Fusion Method for a Quadrotor. *J. Navig.* **2016**, *70*, 325–341. [CrossRef]
27. Baranek, R.; Solc, F. Model-Based Attitude Estimation for Multicopters. *Adv. Electr. Electron. Eng.* **2014**, *12*, 501–510. [CrossRef]
28. Huang, H.; Hoffmann, G.; Waslander, S.; Tomlin, C. Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009; IEEE: Piscataway, NJ, USA, 2010; pp. 3277–3282.
29. Pounds, P.; Mahony, R.; Corke, P. Modelling and Control of a Large Quadrotor Robot. *Control Eng. Pract.* **2010**, *18*, 691–699. [CrossRef]
30. Lyu, P.; Liu, S.; Lai, J.; Liu, J. An analytical fault diagnosis method for yaw estimation of quadrotors. *Control Eng. Pract.* **2019**, *86*, 118–128. [CrossRef]
31. Lyu, P.; Lai, J.; Liu, J.; Liu, H.; Zhang, L. A Thrust Model Aided Fault Diagnosis Method for the Altitude Estimation of a Quadrotor. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *54*, 1008–1019. [CrossRef]
32. Feng, G.; Wu, W.; Wang, J. Observability analysis of a matrix Kalman filter-based navigation system using visual/inertial/magnetic sensors. *Sensors* **2012**, *12*, 8877–8894. [CrossRef] [PubMed]
33. Martinelli, A. State estimation based on the concept of continuous symmetry and observability analysis: The case of calibration. *IEEE Trans. Rob.* **2011**, *27*, 239–255. [CrossRef]
34. Wang, J.; Zhao, M.; Chen, W. MIM_SLAM: A Multi-Level ICP Matching Method for Mobile Robot in Large-Scale and Sparse Scenes. *Appl. Sci.* **2018**, *8*, 2432–2446. [CrossRef]
35. Tian, Y.; Liu, X.; Li, L.; Wang, W. Intensity-Assisted ICP for Fast Registration of 2D-LIDAR. *Sensors* **2019**, *19*, 2124. [CrossRef]