





Article

A Particle Filtering Approach for Fault Detection and Isolation of UAV IMU Sensors: Design, Implementation and Sensitivity Analysis

Egidio D'Amato ¹, Vito Antonio Nardi ^{2,*}, Immacolata Notaro ³ and Valerio Scordamaglia ²

¹ Dipartimento di Scienze e Tecnologie, Università degli Studi di Napoli "Parthenope", 80143 Napoli, Italy; egidio.damato@uniparthenope.it

² Dipartimento di Ingegneria dell'Informazione, delle Infrastrutture e dell'Energia Sostenibile, Università degli Studi "Mediterranea" di Reggio Calabria, 89122 Reggio Calabria, Italy; valerio.scordamaglia@unirc.it

³ Dipartimento di Ingegneria, Università degli Studi della Campania "L. Vanvitelli", 81031 Aversa, Italy; immacolata.notaro@unicampania.it

* Correspondence: vito.nardi@unirc.it

Abstract: Sensor fault detection and isolation (SFDI) is a fundamental topic in unmanned aerial vehicle (UAV) development, where attitude estimation plays a key role in flight control systems and its accuracy is crucial for UAV reliability. In commercial drones with low maximum take-off weights, typical redundant architectures, based on triplex, can represent a strong limitation in UAV payload capabilities. This paper proposes an FDI algorithm for low-cost multi-rotor drones equipped with duplex sensor architecture. Here, attitude estimation involves two 9-DoF inertial measurement units (IMUs) including 3-axis accelerometers, gyroscopes and magnetometers. The SFDI algorithm is based on a particle filter approach to promptly detect and isolate IMU faulted sensors. The algorithm has been implemented on a low-cost embedded platform based on a Raspberry Pi board. Its effectiveness and robustness were proved through experimental tests involving realistic faults on a real tri-rotor aircraft. A sensitivity analysis was carried out on the main algorithm parameters in order to find a trade-off between performance, computational burden and reliability.

Keywords: fault detection and isolation; particle filter; UAV fault detection; FDI; fault tolerant attitude estimation; duplex attitude estimation architecture; analytical redundancy



Citation: D'Amato, E.; Nardi, V.A.; Notaro, I.; Scordamaglia, V. A Particle Filtering Approach for Fault Detection and Isolation of UAV IMU Sensors: Design, Implementation and Sensitivity Analysis. *Sensors* **2021**, *21*, 3066. <https://doi.org/10.3390/s21093066>

Academic Editor: Piotr Witczak

Received: 31 March 2021

Accepted: 25 April 2021

Published: 28 April 2021

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



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years the use of unmanned aircraft has significantly increased, making the mitigation of risks due to on-board avionics malfunctions essential, in view of sustainable and safe aviation.

Several applications have been exploited with military and civil objectives [1,2], including search and rescue missions in hostile environments [3], surveillance and security patrols [4], home security [5], precision agriculture [4] and network coverage [6], fields of application where UAVs play a key role in terms of reduction in operations and support costs [4].

The *flight control system* (FCS) is a key component of a generic UAV. It can include inertial sensors, air data systems, avionics, control surfaces/servos, servo-actuators, on-board software and other relevant subsystems contributing to UAV stability and control. A fault in such a system could be critical in the assessment of UAV reliability. It is worth noting that faults on FCSs account for 25% of U.S. military UAV incidents [7].

An FDI system must be able to detect and properly isolate faults in order to mitigate the effects on UAV system performance and integrity. Four key features should be present in an FDI system:

- fast detection of abnormal situations;

- isolation of faults;
- robustness to noise and uncertainties;
- low false alarm rate.

For a modern aircraft, three kinds of fault can be considered:

- sensor faults: a malfunction of the measurement subsystem;
- actuator faults: a malfunction of actuators acting on system dynamics;
- process faults: a strong change in system dynamics due to structural problems.

The most sophisticated UAVs are often equipped with technology solutions for automatically detecting and isolating faults on sensors, collectively labeled as *fault detection and isolation* (FDI) systems [8–12]. The presence of an FDI system increases the capability of UAVs to fulfill prescribed tasks even in the presence of faults. Quick detection and proper isolation of faults represents the first step of a reconfiguration strategy, required to minimize fault impact upon UAV performance [9,13]. In the literature, significant research efforts have been spent in terms of FDI solutions for UAVs. In [14], FDI on UAV Pitot tubes and control surfaces is carried out by using statistical change detection, while in [15], a state machine-based solution is proposed. Ref. [16] shows a sensor fusion-approach for FDI on air data sensors and [17] exploits self-tuning residual generators for fault detection on control surfaces. Finally, a cooperative FDI scheme for UAVs is proposed in [18,19] deals with the faults in UAV FDI systems.

In order to control UAV orientation with respect to an inertial reference system, one of the FCS tasks is attitude estimation. Usually, this consists of sensor fusion algorithms that use measurements coming from accelerometers, gyroscopes and magnetometers [20]. Being an essential task, the problem of UAV attitude estimation on the basis of sensor measurements can be non-trivial [21], and sensor fault detection must be considered to minimize risks due to loss of control.

In aerospace and several industrial applications, hardware redundancy-based techniques are the most common solutions to tackle faults on sensors. Such solutions require a set of redundant sensors to validate measurement data [22]. A typical approach in commercial and military aircraft is based on triple or more hardware redundancies to enhance safety and reliability [23]. Although hardware redundancy techniques are convenient solutions from the point of view of implementation and management, they bring several drawbacks such as weight, power consumption and costs, which are all relevant in small-size UAV applications.

In the literature, some effort has been spent trying to replace hardware redundancy-based techniques with analytical redundancy-based ones [23–32]. In [25] a Kalman filtering approach is implemented in order to reveal oscillatory failures of redundant aircraft sensors, used for flight control law calculation. An FDI method that exploits set-valued observers is proposed in [26] for uncertain linear parameter-varying systems. This method uses a bank of filters and it does not need thresholds to state a fault, in contrast with residual-based architectures. A fault-tolerant scheme for the sensors of a large civil aircraft is proposed in [28,29] using Kalman filters and sliding mode observers. Further nonlinear and robust techniques, such as H_∞ filtering, can reinforce robustness to faults of physically redundant schemes [32,33].

In recent years, some FDI solutions based on particle filters (PFs) are also proposed. A PF is a sequential Monte Carlo method (SMC)[34] aimed at estimating the state of a dynamical system subject to random perturbations, via noisy output observations. A particle filter solution represents an attractive choice for UAV attitude estimation problems [20,35–39] outperforming classical Kalman filter (KF)-based approaches [40] at the price of significant computational complexity growth [20]. A PF-based fault detection scheme is proposed by [41], while [42] highlights PF advantages in tackling FDI for non-linear systems and [43] also discusses PF drawbacks. In [44] a particle filter deals with the FDI problem for the autonomous integrity monitoring of a GPS receiver. Ref. [45] proposes two different PF-based FDI architectures, which are compared with more classical KF-based approaches. In [46] a particle filter solution is used to deal with both known and unknown faults for a

complex system, focusing on a wheeled mobile robot, and in [47] a PF replaces principal component analysis in driving a Gaussian Mixture Model for process monitoring FDI. In [48] a general PF-based FDI scheme is proposed. Similarly [49] is focused upon tackling some of the most well-known PF problems.

In commercial unmanned aviation, costs are a key objective to ensure that UAVs become widespread and, in specific operations, risk assessment needs proof that the flight risk can be acceptable.

In such direction, this paper is focused on the implementation of a sensor fault detection and isolation technique on low-cost hardware, typical of commercial micro- and mini-UAVs. The scope of the proposed procedure is to detect and isolate faults on inertial measurement units used to estimate UAV attitude. The considered hardware is based on a duplex architecture with two inertial measurement units (IMUs) including 3-axis gyroscopes, accelerometers and magnetometers. The estimation problem is dealt with by the design of a particle-filter-based sensor fusion algorithm. From a theoretical point of view, a PF could be a realistic option when statistical performance is considered [50–52], because it is able to deal with non-linear motion models and non-Gaussian noise distributions.

Although the use of a PF approach [20] can increase the computational burden, the paper shows promising results by using an SFDI procedure based on two parallel PFs.

The effectiveness of the proposed procedure to detect and isolate IMU sensor faults is proved by means of numerical simulations conducted on a Raspberry Pi board using real data gathered during the flight of a tri-rotor aircraft. Furthermore, this work proposes a sensitivity analysis involving main algorithm parameter variations to estimate effects in terms of SFDI performance. The computation time on a low-cost and low-power embedded platform is assessed in order to prove the algorithm's applicability in a real-world scenario.

The paper is organized as follows:

- in Section 2, the proposed hardware and software architecture are described;
- in Section 3, the attitude estimation model is defined, using quaternion-based relations;
- in Section 4, the particle filter algorithm is introduced;
- in Section 5, the SFDI algorithm is shown with details on the implementation;
- the effectiveness of the algorithm is shown by means of realistic simulations in Section 6, where an analysis against tunable parameters is discussed.

2. System Description

The proposed hardware architecture is composed of the following components:

- two commercially available boards based on the STM32F103VCT CPU by ST, typically used on low-cost multi-rotors as flight controllers; such boards, called IMU-1 and IMU-2, weighing 17 grams, have a power consumption average below 1W and include the following inertial sensors:
 - an MPU6050-based MEMS triaxial accelerometer and gyroscope (see specifications in Table 1);
 - an HMC5883L-based MEMS magnetometer (see specifications in Table 1);
- a Raspberry Pi 3B platform, a single-board computer based on the Broadcom BCM2837 64bit CPU, with wireless LAN, USB and GPIO connectivity. It weighs 42 grams and in our test had a power consumption of less than 2.5W.

IMU-1 and IMU-2 are linked to the Raspberry Pi platform by USB ports through an on-board *serial to USB* chip. This configuration simplified the prototyping while ensuring a high bandwidth for data transmission.

Figure 1 summarizes the structure of the experimental setup.

IMUs act as data collectors. The on-board software acquires signals from sensors on the I²C port at a frequency of 100 Hz and sends them through a serial connection.

The SFDI algorithm, together with particle filtering, was implemented on the Raspberry Pi 3B platform as a real-time task over a Debian-based operating system with a fully preemptive GNU/Linux 4.14 kernel. It was coded using MATLAB R2019b by Mathworks

and was made executable on the embedded hardware by using code generation. The produced C code was properly modified to customize peripherals access and recompiled on the board. During our tests, the SFDI executable had the highest execution priority to minimize any jitter due to the operating system.

The goal of the SFDI algorithm is to monitor the health of each IMU and to supervise the overall system by providing reliable data to the flight control system. In standard conditions, the output of the SFDI algorithm is the average attitude estimated using both IMUs, while, when a fault is detected, the procedure provides as outputs only the orientation computed by the operational IMU.

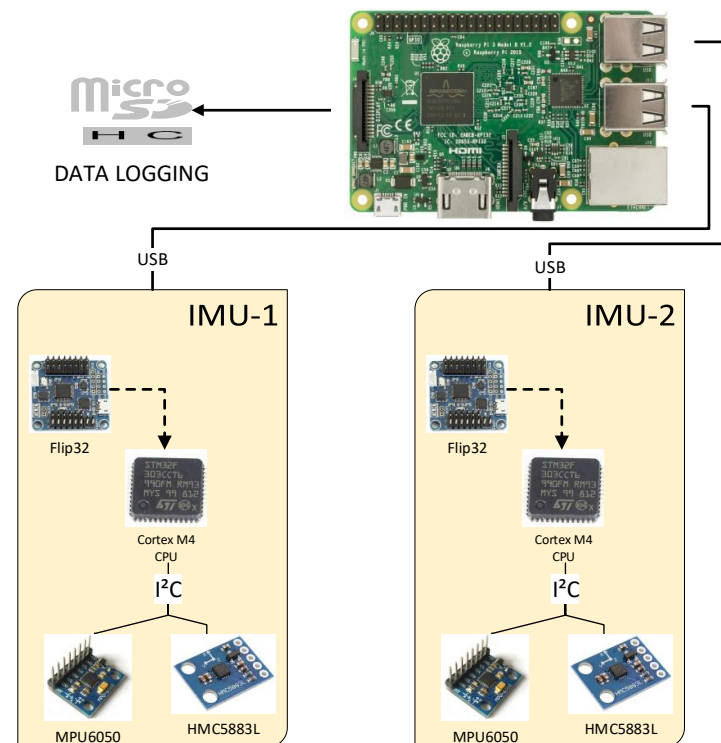


Figure 1. Architecture of the proposed duplex IMU. On top, the Raspberry Pi 3B platform, running the SFDI algorithm and data logging. Below, Flip32-based IMU boards, with MPU6050 and HMC5883L MEMS sensors.

Table 1. MEMS sensor characteristics from datasheets.

Name	MPU6050 Acc.	MPU6050 Gyro	HMC58831 Mag.
Dynamic range	± 4 g	± 500 deg s ⁻¹	± 2.5 Gs
Digital resolution	0.122 mg	0.015 deg s ⁻¹	2.095 mGs
Total RMS noise	n.d.	0.05 deg s ⁻¹	n.d.
Noise density	400 μgHz^{-2}	0.005 deg s ⁻¹ Hz ⁻²	n.d.
Noise floor	n.d.	n.d.	0.002 mGs

3. UAV Attitude Estimation Problem

Vehicle attitude is defined as its orientation with respect to a reference frame and its accurate estimation is an important topic in the robotics and aerospace fields. UAV attitude is obtained by using a sensor fusion algorithm, together with data provided by IMU sensors.

The attitude obtained by integrating gyro measurements in terms of angular velocities is affected by non-negligible issues, due to bias and random-walk errors. On the other hand, accelerometers and magnetometers provide a good estimate only in static or quasi-static conditions, in the absence of any constant acceleration and magnetic field changes. In order to mitigate such problems, different sensor data fusion algorithms can be implemented

on-board [53,54]. Most UAV attitude estimation algorithms are based on an extended Kalman filter [20,55–57], unscented Kalman filter [58,59], particle filtering [36,51,60], quaternion estimation (QUEST) algorithms [61,62], complementary filter [63,64] and Madgwick filter [65].

To describe the attitude estimation problem, two reference systems must be defined:

- a north-east-down (NED) reference frame, parallel to the Earth's surface, with the X_E -axis pointed toward north, the Y_E -axis pointed toward east and the Z_E -axis oriented downwards;
- the body frame, centered in the UAV's center of gravity (CG), with the X_B -axis pointed toward the UAV nose, the Z_B -axis downward and the Y_B -axis oriented to complete a right-handed system.

Since UAV navigation is done in a bounded space, the NED frame can be centered in the UAV departing point and assumed as an inertial reference system.

Transition from the inertial to the body frame is obtained with a sequence of three ordered rotations, denoted as 3 – 2 – 1 [66] (see Figure 2):

- the first rotation around the Z_E axis by the yaw angle ψ , from $OX_EY_EZ_E$ to $OX'Y'Z'$;
- the second rotation around the Y' axis by the pitch angle θ from $OX'Y'Z'$ to $OX''Y''Z''$;
- the third rotation around the X'' axis by the roll angle ϕ from $OX''Y''Z''$ to $OX_BY_BZ_B$.

The angles ψ , θ and ϕ are called Euler angles and describe the aircraft's attitude, i.e., the orientation of the body frame with respect to the inertial NED frame.

Alternatively, to define the relative rotation of the body frame with respect to the inertial frame, a quaternion-based representation can be used [67]. From Euler's theorem, any rotation sequence is equivalent to a single rotation by a given angle α about a fixed axis passing through the origin. This rotation can be described by the unit quaternion vector $\tilde{\mathbf{q}} = [\tilde{q}_0, \tilde{q}_1, \tilde{q}_2, \tilde{q}_3]^T$, whose components define the axis vector $\boldsymbol{\zeta} = [\zeta_x, \zeta_y, \zeta_z]^T$ and the angle α (see Figure 3):

$$\begin{aligned}\tilde{q}_1 &= \zeta_x, & \tilde{q}_2 &= \zeta_y, & \tilde{q}_3 &= \zeta_z, & \tilde{q}_0 &= \cos \alpha \\ |\tilde{\mathbf{q}}| &= \sqrt{\tilde{q}_0^2 + \tilde{q}_1^2 + \tilde{q}_2^2 + \tilde{q}_3^2}\end{aligned}$$

Let us call $\mathbf{q} = [q_0, q_1, q_2, q_3]^T$ the unit quaternion vector associated to $\mathbf{R}_{BE}(\mathbf{q})$ from the NED reference frame to the body frame:

$$\mathbf{R}_{BE}(\mathbf{q}) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_0^2 & 2(q_1q_2 - q_3q_0) & 2(q_1q_3 + q_2q_0) \\ 2(q_1q_2 + q_3q_0) & -q_1^2 + q_2^2 - q_3^2 - q_0^2 & 2(q_2q_3 - q_1q_0) \\ 2(q_1q_3 - q_2q_0) & 2(q_2q_3 + q_1q_0) & -q_1^2 - q_2^2 + q_3^2 + q_0^2 \end{bmatrix} \quad (1)$$

Euler angle-based representation and quaternion-based representation are equivalent; therefore, a one-to-one correspondence between Euler angles and the quaternion vector exists:

$$\mathbf{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} + \sin \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \sin \frac{\phi}{2} \cos \frac{\theta}{2} \cos \frac{\psi}{2} - \cos \frac{\phi}{2} \sin \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} - \sin \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} \\ \cos \frac{\phi}{2} \cos \frac{\theta}{2} \sin \frac{\psi}{2} - \sin \frac{\phi}{2} \sin \frac{\theta}{2} \cos \frac{\psi}{2} \end{bmatrix} \quad (2)$$

The Euler angles are computed as follows [66]:

$$\phi = \arctan \left(\frac{2q_0q_1 + 2q_2q_3}{1 - 2q_1^2 - 2q_2^2} \right) \tag{3}$$

$$\theta = \arcsin (2q_0q_2 - 2q_3q_1) \tag{4}$$

$$\psi = \arctan \left(\frac{2q_0q_3 + 2q_1q_2}{1 - 2q_2^2 - 2q_3^2} \right) \tag{5}$$

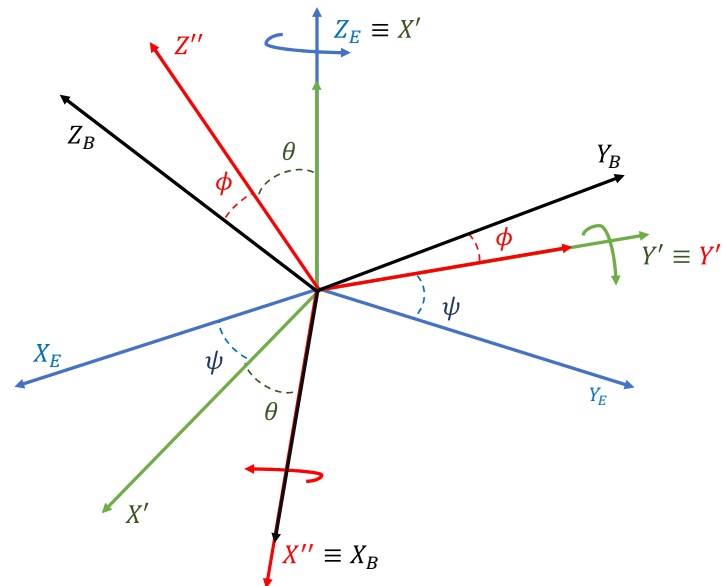


Figure 2. Inertial (E) and body-fixed (B) reference frames. Transition from the NED frame to the body frame is obtained with a sequence of three ordered rotations around the Z_E axis, the Y' axis and the X'' axis of angles ψ , θ and ϕ , respectively.

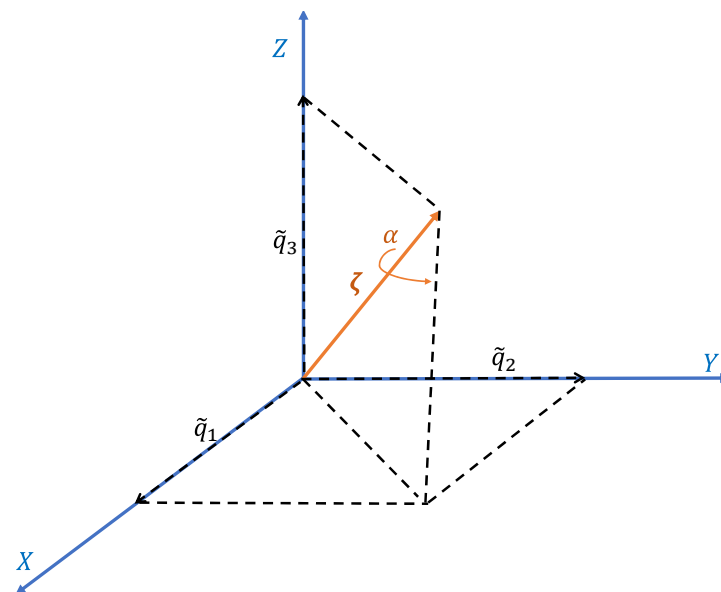


Figure 3. Geometrical meaning of quaternion: a rotation around the ζ axis by a given angle α is defined by the unit quaternion $\tilde{\mathbf{q}} = [\tilde{q}_0, \tilde{q}_1, \tilde{q}_2, \tilde{q}_3]^T$, whose components are $\tilde{q}_1 = \zeta_x$, $\tilde{q}_2 = \zeta_y$, $\tilde{q}_3 = \zeta_z$ and $\tilde{q}_0 = \cos \alpha$.

In the absence of noise and uncertainties in IMU measurements, the UAV attitude estimation could be carried out by integrating the angular speed vector $\boldsymbol{\omega} = [p, q, r]^T$ from a known initial attitude. Consequently, the quaternion vector exhibits the following dynamics:

$$\dot{\mathbf{q}}(t) = \frac{1}{2} \mathbf{Q}(\boldsymbol{\omega}(t)) \cdot \mathbf{q}(t) \quad (6)$$

where the matrix $\mathbf{Q}(\boldsymbol{\omega}(t))$ is

$$\mathbf{Q}(\boldsymbol{\omega}(t)) = \begin{bmatrix} 0 & -p(t) & -q(t) & -r(t) \\ p(t) & 0 & r(t) & -q(t) \\ q(t) & -r(t) & 0 & p(t) \\ r(t) & q(t) & -p(t) & 0 \end{bmatrix} \quad (7)$$

However, the gyroscope output is an angular velocity affected by white noise. In static conditions, in the absence of rotations, the gyroscope output is not zero as expected, but it is a white noise with a zero mean and a given standard deviation. On a finite time horizon, the integration will lead to a drifting of the angle estimation [68,69].

The sensed rotational velocities $\boldsymbol{\omega}_S(t)$ can be assumed different from $\boldsymbol{\omega}$ due to the presence of bias and stochastic noise:

$$\boldsymbol{\omega}_S(t) = \begin{bmatrix} p_s(t) \\ q_s(t) \\ r_s(t) \end{bmatrix} = \boldsymbol{\omega}(t) + \mathbf{b}(t) + \mathbf{d}(t) \quad (8)$$

$\mathbf{b}(t) = [b_p(t), b_q(t), b_r(t)]^T$ and $\mathbf{d}(t) = [d_p(t), d_q(t), d_r(t)]^T$ represent the vectors of biases and unknown zero-mean noise affecting gyroscopes, respectively.

In steady-state and slowly varying conditions, a good solution is to involve accelerometers and magnetometers, being dependent on the UAV orientation. The relationship between UAV orientation and accelerometer and magnetometer output vectors \mathbf{a}^B and \mathbf{M}^B is defined as follows:

$$\mathbf{a}^B = \mathbf{R}_{BE}(\mathbf{q}) \cdot \mathbf{a}^E \quad (9)$$

$$\mathbf{M}^B = \mathbf{R}_{BE}(\mathbf{q}) \cdot \mathbf{M}^E \quad (10)$$

where $\mathbf{g}^E = [0 \ 0 \ g]^T$ is the gravity vector with $g = 9.801 \text{ m/s}^2$ and \mathbf{M}^E is the Earth's magnetic field, both defined in the Earth frame.

However, although an optimization-based procedure can be used to solve Equations (9) and (10), the last approach is not reliable [67,70], due to the presence of non-negligible noise in measurements.

In order to exploit (9) and (10), consider the sensed accelerations and field magnetic measurements:

$$\mathbf{a}_S(t) = \begin{bmatrix} a_{x,S}(t) \\ a_{y,S}(t) \\ a_{z,S}(t) \end{bmatrix} = \mathbf{a}^B(t) + \mathbf{v}_a(t) \quad (11)$$

$$\mathbf{M}_S(t) = \begin{bmatrix} M_{x,S}(t) \\ M_{y,S}(t) \\ M_{z,S}(t) \end{bmatrix} = \mathbf{M}^B(t) + \mathbf{v}_M(t) \quad (12)$$

with $\mathbf{v}_a(t)$ and $\mathbf{v}_M(t)$ vectors of sensor noise affecting triaxial accelerometers and magnetometers at the time instant t , respectively.

The attitude estimation can be carried out by merging both approaches [20,71]. The kinematic model (6) is usually extended with additional dynamics in order to take into account the gyroscope biases [72]:

$$\dot{\mathbf{q}}(t) = \frac{1}{2} \mathbf{Q}(\boldsymbol{\omega}_S(t) - \mathbf{b}(t) - \mathbf{d}(t)) \cdot \mathbf{q}(t) \quad (13)$$

$$\dot{\mathbf{b}}(t) = -\frac{1}{\tau} \mathbf{b}(t) \quad (14)$$

where τ is a time constant compliant with bias dynamics. To correct the previous estimation, (15) and (16) can be used:

$$\mathbf{a}_S(t) = \mathbf{R}_{BE}(\mathbf{q}(t)) \cdot \mathbf{g}^E + \mathbf{v}_a(t) \quad (15)$$

$$\mathbf{M}_S(t) = \mathbf{R}_{BE}(\mathbf{q}(t)) \cdot \mathbf{M}^E + \mathbf{v}_M(t) \quad (16)$$

Equations (13)–(16) can be expressed in the following generic form:

$$\dot{\mathbf{x}}(t) = \boldsymbol{\gamma}(\mathbf{x}(t), \boldsymbol{\omega}_S(t), \mathbf{d}(t)) \quad (17)$$

$$\mathbf{y}(t) = \boldsymbol{\lambda}(\mathbf{x}(t), \mathbf{v}(t)) \quad (18)$$

where $\mathbf{x}(t) = [\mathbf{q}(t)^T \quad \mathbf{b}(t)^T]^T$ is the state vector, $\mathbf{y}(t) = [\mathbf{a}_S(t)^T \quad \mathbf{M}_S(t)^T]^T$ is the output vector and $\mathbf{v}(t) = [\mathbf{v}_a(t)^T \quad \mathbf{v}_M(t)^T]^T$ is the noise vector.

In such representation, accelerometers and magnetometers provide output sensor measurements according to (18), whereas gyroscopes represent an external input acting on (17).

The attitude estimation problem can be reformulated as the estimation of the state vector \mathbf{x} from (17) based on output measurements (18). Corrective actions can then be designed and implemented, feeding back the error between measured and estimated outputs. Optimal estimation has the role of reducing sensitivity to noise and forcing the additional state vector components \mathbf{b} to compensate for gyro biases or other low-frequency uncertain contributions.

4. Particle Filter Algorithm

Let us assume the following generic discrete time representation:

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), \boldsymbol{\omega}_S(k-1), \mathbf{d}(k-1)) \quad (19)$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k)) + \mathbf{v}(k) \quad (20)$$

being $\mathbf{x} \in \mathcal{R}^{n_x}$, $\mathbf{y} \in \mathcal{R}^{n_y}$, $\boldsymbol{\omega}_S \in \mathcal{R}^{n_\omega}$, $\mathbf{v} \in \mathcal{R}^{n_v}$ and $\mathbf{d} \in \mathcal{R}^{n_d}$. It can be obtained by applying Euler's discretization to (17) and (18) where $\mathbf{x}(k) = \mathbf{x}(t_k)$, $\mathbf{y}(k) = \mathbf{y}(t_k)$, $\boldsymbol{\omega}_S(k) = \boldsymbol{\omega}_S(t_k)$, $\mathbf{d}(k) = \mathbf{d}(t_k)$ and $\mathbf{v}(k) = \mathbf{v}(t_k)$ and t_k the k -th sampling time instant

$$t_k = \sum_{i=0}^{k-1} T_S \quad \forall k \geq 1, \quad t_0 = 0 \quad (21)$$

with $T_S > 0$ the constant sampling time.

Equations (19) and (20) represent a partially observed Markov process, where the output vector $\mathbf{y}(\cdot)$ is observable, while the state $\mathbf{x}(\cdot)$ is unobservable. A Markov process is a stochastic process whose state vector $\mathbf{x}(k)$ depends only on $\mathbf{x}(k-1)$, being independent of $\mathbf{x}(k-l)$, with $l > 1$.

A particle filter is used to estimate $\mathbf{x}(k)$ in (19), fulfilling (20) under the hypothesis that the process $\mathbf{d}(k-1)$ and the measurement noise $\mathbf{v}(k)$ are unknown.

The particle filter is a numeric implementation of a Bayesian estimator [73]. The Bayesian approaches are based on Bayes' rule. Their goal is to estimate the conditional

probability density function (PDF) of the current state $\mathbf{x}(k)$, given the whole set of acquired measurements at time k , denoted by $\mathcal{Y}(k) = \{\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(k)\}$. The conditional PDF is denoted as $p(\mathbf{x}(k)|\mathcal{Y}(k))$.

The Bayesian estimator can be formulated in a recursive way, in order to update the PDF when a new measurement is acquired [74].

At the first time step $k = 0$, since no measurements are available, e.g., the set $\mathcal{Y}(0) = \emptyset$, the estimator is initialized as follows:

$$p(\mathbf{x}(0)|\mathcal{Y}(0)) = p(\mathbf{x}(0)) \quad (22)$$

where $p(\mathbf{x}(0))$ is the known initial PDF of state.

At each time step k , the a priori PDF denoted by $p(\mathbf{x}(k)|\mathcal{Y}(k-1))$ is computed using the Chapman–Kolmogorov formula [73]:

$$p(\mathbf{x}(k)|\mathcal{Y}(k-1)) = \int p(\mathbf{x}(k)|\mathbf{x}(k-1)) \cdot p(\mathbf{x}(k-1)|\mathcal{Y}(k-1)) \cdot d\mathbf{x}(k-1) \quad (23)$$

In the a posteriori phase, the conditional PDF is computed by using Bayes' rule as follows:

$$p(\mathbf{x}(k)|\mathcal{Y}(k)) = \frac{p(\mathbf{y}(k)|\mathbf{x}(k)) \cdot p(\mathbf{x}(k)|\mathcal{Y}(k-1))}{p(\mathbf{y}(k)|\mathcal{Y}(k-1))} \quad (24)$$

where $p(\mathbf{y}(k)|\mathbf{x}(k))$ is available thanks to measurement equations and the PDF of the measurement noise $\mathbf{v}(k)$.

The PDF $p(\mathbf{y}(k)|\mathcal{Y}(k-1))$ depends on the a priori PDF $p(\mathbf{x}(k)|\mathcal{Y}(k-1))$ and $p(\mathbf{y}(k)|\mathbf{x}(k))$:

$$p(\mathbf{y}(k)|\mathcal{Y}(k-1)) = \int p(\mathbf{x}(k)|\mathbf{x}(k-1)) \cdot p(\mathbf{x}(k-1)|\mathcal{Y}(k-1)) \cdot d\mathbf{x}(k-1) \quad (25)$$

Equations (23)–(25) involve intractable multidimensional integrals [34]. The particle filtering approach offers a numerical solution to this problem [75].

In PF algorithms, PDF $p(\mathbf{x}(k)|\mathcal{Y}(k))$ can be approximated by a set of random samples named *particles*. According to [76], the convergence rate of the particle-approximated cumulative function is $O\left(\frac{1}{\sqrt{\Delta}}\right)$, with Δ a constant tunable parameter indicating the number of particles.

An important feature that characterizes particle filter algorithms is the method used to obtain new samples. A widely used class of resampling techniques is *sequence importance resampling* (SIR) [77].

Let us assume discrete-time model (19) and (20). A particle filter is used to estimate the $\mathbf{x}(k)$ value knowing the vector $\mathbf{y}(k)$ of accelerometer and magnetometer measurements and the vector $\boldsymbol{\omega}_S(k-1)$ of the rotational speed sensed by the gyroscopes.

Assuming that the PDF of the initial state $p(\mathbf{x}(0))$ is known, the PF algorithm is initialized randomly generating Δ particles $\hat{\mathbf{x}}(0)$, on the basis of PDF $p(\mathbf{x}(0))$. Then, at each time step k , the procedure involves the following steps:

- (i) the set of a priori particles $\mathbf{X}(k|k-1)$ at the k -th sampling step is obtained by propagating the set $\mathbf{X}(k-1|k-1)$ with the state Equation (19), considering the process noise $\mathbf{d}(k)$ equal to zero. Therefore, the generic particle $\mathbf{x}_\delta(k) \in \mathbf{X}(k|k-1)$ is computed as follows:

$$\mathbf{x}_\delta(k) = \mathbf{f}(\hat{\mathbf{x}}_\delta(k-1), \boldsymbol{\omega}_S(k-1), \mathbf{0}) \quad (26)$$

being $\hat{\mathbf{x}}_\delta(k-1) \in \mathbf{X}(k-1|k-1)$ with $\delta = 1, \dots, \Delta$.

- (ii) We compute the relative likelihood $\tilde{w}_\delta(k)$ of each particle $\mathbf{x}_\delta(k)$ conditioned on the measurement $\mathbf{y}(k)$, i.e., evaluate the PDF $p(\mathbf{y}(k)|\mathbf{x}_\delta(k))$ on the basis of the nonlinear measurement equation and the PDF of the measurement noise $\mathbf{v}(k)$. Under the

assumption of additive measurement noise, the weight $\tilde{w}_\delta(k)$ can be computed as follows:

$$\tilde{w}_\delta(k) = P[\mathbf{y}(k)|\mathbf{x}_\delta(k)] = P[\mathbf{v}(k) = (\mathbf{y}(k) - \mathbf{y}_\delta(k))] \quad (27)$$

where $P(\cdot)$ denotes the probability and \mathbf{y}_δ is obtained according to the output equation, neglecting the measurement noise $\mathbf{v}(k)$:

$$\mathbf{y}_\delta(k) = \mathbf{h}(\mathbf{x}_\delta(k)) \quad (28)$$

(iii) The weights $\tilde{w}_\delta(k)$ are normalized according to the following equation:

$$w_\delta(k) = \frac{\tilde{w}_\delta(k)}{\sum_{\delta=1}^{\Delta} \tilde{w}_\delta(k)} \quad (29)$$

(iv) A set of particles taking into account a posteriori knowledge $\tilde{\mathbf{X}}(k|k)$ is obtained by randomly drawing Δ samples from $\mathbf{X}(k|k-1)$

$$\tilde{\mathbf{X}}(k|k) = \left\{ \underbrace{\mathbf{x}_\delta(k) \in \mathbf{X}(k|k-1), \mathbf{x}_1(k), \dots, \mathbf{x}_1(k), \dots}_{m_1}, \dots, \underbrace{\mathbf{x}_\delta(k), \dots, \mathbf{x}_\delta(k), \dots, \mathbf{x}_\Delta(k), \dots, \mathbf{x}_\Delta(k)}_{m_\Delta} \right\} \quad (30)$$

therefore, each particle $\mathbf{x}_\delta \in \mathbf{X}(k|k-1)$ is chosen m_δ times. The integer number m_δ must satisfy the following conditions:

- m_δ is a non-negative integer;
- $\sum_{\delta=1}^{\Delta} m_\delta = \Delta$;
- $\lim_{\Delta \rightarrow \infty} m_\delta = w_\delta(k) \cdot \Delta$.

To compute the parameter m_δ the following sub-procedure is applied:

(a) define a series of $\Delta + 1$ thresholds σ_δ as:

$$\sigma_\delta = \sum_1^\delta w_\delta, \quad \text{with } \sigma_0 = 0 \quad (31)$$

- (b) initialize m_δ values to zero;
- (c) draw Δ uniformly distributed random numbers r_i from the interval $[0, 1]$, $\mathcal{P} = \{r_i \in [0, 1], \text{ with } i = 1, \dots, \Delta\}$;
- (d) for each $r_i \in \mathcal{P}$, find δ such that $\sigma_{\delta-1} < r_i \leq \sigma_\delta$, then $m_\delta = m_\delta + 1$.
- (v) In order to counteract the so-called *particle degeneracy problem*, which can weaken algorithm convergence and robustness [78], the set of a posteriori particle $\mathbf{X}(k|k)$ is obtained by randomly scattering each particle of set $\tilde{\mathbf{X}}(k|k)$ in a given neighborhood.
- (vi) The state estimate $\hat{\mathbf{x}}(k)$ is evaluated as the algebraic mean of a posteriori particles belonging to $\mathbf{X}(k|k)$:

$$\hat{\mathbf{x}}(k) = \frac{1}{\Delta} \sum_{\delta=1}^{\Delta} \hat{\mathbf{x}}_\delta(k) \quad (32)$$

In Section 6.2 a sensitivity analysis against introduced tunable parameters Δ and ν is proposed.

5. SFDI Algorithm for a Duplex IMU

In the view of accomplishing navigation purposes, let us consider an onboard flight controller based on a *duplex sensor architecture* including two IMUs, named IMU-1 and IMU-2, with triaxial accelerometers, gyroscopes and magnetometers, whose output sensor measurements $\mathbf{y}^{(1)}(k)$ and $\mathbf{y}^{(2)}(k)$ are defined as follows:

$$\mathbf{y}^{(i)}(k) = \begin{bmatrix} \mathbf{a}_S^{(i)}(k) \\ \mathbf{M}_S^{(i)}(k) \end{bmatrix} \quad i = 1, 2 \quad (33)$$

where $\mathbf{a}_S^{(i)}(k)$, $\mathbf{M}_S^{(i)}(k)$ are the outputs of the i -th triaxial accelerometer and magnetometer at the time instant k .

According to (8), the gyroscope measurements $\omega_S^{(i)}(k)$ are used as the input of the kinematic model at the time instant k .

For each IMU i , the quaternion vector $\mathbf{q}^{(i)}(k)$ and gyroscope biases $\mathbf{b}^{(i)}(k)$ are estimated by using a particle filter, named *PFi*, fed with $\omega_S^{(i)}(k)$ and $\mathbf{y}^{(i)}(k)$.

A sensor reconfiguration logic has been implemented to achieve FDIR, formulated as an *event-driven* state machine, based on the following sets of logical states:

$$\chi(k) = [N_A(k), A_A(k), N_G(k), A_G(k), N_M(k), A_M(k), F_1(k), F_2(k), R_1(k), R_2(k)] \quad (34)$$

where

- subscript A indicates accelerometers, subscript G is for gyroscopes and subscript M is for magnetometers;
- N_Z is the normal state for sensor $Z \in \{A, G, M\}$;
- A_Z is the alert state activated at the detection phase on the basis of a preliminary comparison between IMUs;
- F_j is the fault state for the IMU unit $j \in \{1, 2\}$.

In Figure 4, the state transition graph is shown. The events driving the state transition of the proposed state machine can be grouped into three algorithm steps:

Step 1—FAULT DETECTION. Under the assumption of a non-contemporary fault, at each time step k , the transition from normal $N(k-1)$ to alert $A(k)$ state (and vice versa from $A(k-1)$ to $N(k)$) is achieved by comparing measurement accelerometer, magnetometer and gyroscope measurements from IMUs.

$$\mathcal{E}_{A1} \Leftrightarrow \exists l \in \{1, 2, 3\} : |a_{l,S}^{(1)}(k) - a_{l,S}^{(2)}(k)| \geq \tau_a \quad (35)$$

$$\mathcal{E}_{A2} \Leftrightarrow \forall l \in \{1, 2, 3\}, |a_{l,S}^{(1)}(k) - a_{l,S}^{(2)}(k)| < \tau_a \quad (36)$$

$$\mathcal{E}_{M1} \Leftrightarrow \exists l \in \{1, 2, 3\} : |M_{l,S}^{(1)}(k) - M_{l,S}^{(2)}(k)| \geq \tau_m \quad (37)$$

$$\mathcal{E}_{M2} \Leftrightarrow \forall l \in \{1, 2, 3\}, |M_{l,S}^{(1)}(k) - M_{l,S}^{(2)}(k)| < \tau_m \quad (38)$$

$$\mathcal{E}_{G1} \Leftrightarrow \exists l \in \{1, 2, 3\} : |\omega_{l,S}^{(1)}(k) - \omega_{l,S}^{(2)}(k)| \geq \tau_g \quad (39)$$

$$\mathcal{E}_{G2} \Leftrightarrow \forall l \in \{1, 2, 3\}, |\omega_{l,S}^{(1)}(k) - \omega_{l,S}^{(2)}(k)| < \tau_g \quad (40)$$

where $a_{l,S}$, $M_{l,S}$ and $\omega_{l,S}$ are the l -th components of accelerometer, magnetometer and gyroscope measurements, respectively. τ_a , τ_m and τ_g are positive scalar thresholds for accelerometers, magnetometers and gyroscopes, required to make the detection robust against sensor noise and uncertainties.

Step 2—FAULT ISOLATION. Once the presence of a fault is detected, with \tilde{k} the time instant of fault detection, then

$$\mathbf{b}_0^{(i)} = \mathbf{b}^{(i)}(\tilde{k})$$

As for the transition from the alert to the faulted state, a fault on the i -th IMU is declared if the following condition holds:

$$\mathcal{E}_3^i \Leftrightarrow \|\Omega \cdot (\mathbf{q}^{(1)}(k) - \mathbf{q}^{(2)}(k))\|_\infty > 1 \quad (41)$$

where $\|\cdot\|_\infty$ denotes the infinity norm and Ω is a diagonal weighing matrix required to take into account the effects of sensor noise. The faulted IMU \bar{i} can be isolated solving the following problem:

$$\bar{i} = \arg \max_{i=1,2} \sigma(i) \quad (42)$$

The residual $\sigma(i)$ represents the error between output estimation and measurements, increased with the difference between the estimated biases at the current time and at the detection time:

$$\sigma(i) = \left\| \begin{bmatrix} \mathbf{y}^{(i)}(k) \\ \mathbf{b}_0^{(i)} \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{y}}^{(i)}(k) \\ \hat{\mathbf{b}}^{(i)}(k) \end{bmatrix} \right\|_2 \quad i = 1, 2 \quad (43)$$

where $\|\cdot\|_2$ indicates the Euclidean norm.

According to (42), a faulted sensor belongs to the IMU whose output is farther from the relevant full particle filter forecast.

Step 3—FAULT RECOVERY. When the fault condition is no longer held (\mathcal{E}_{Z2}), the system is able to recover the faulted IMU by checking the consistency between attitude estimations from both *PFs*, through

$$\mathcal{E}_4^{\bar{i}} \Leftrightarrow \|\Omega \cdot (\mathbf{q}^{(1)}(k) - \mathbf{q}^{(2)}(k))\|_\infty < 1 \quad (44)$$

In Section 6.2 a sensitivity analysis against introduced tunable parameters τ_a , τ_m , τ_g and Ω is proposed.

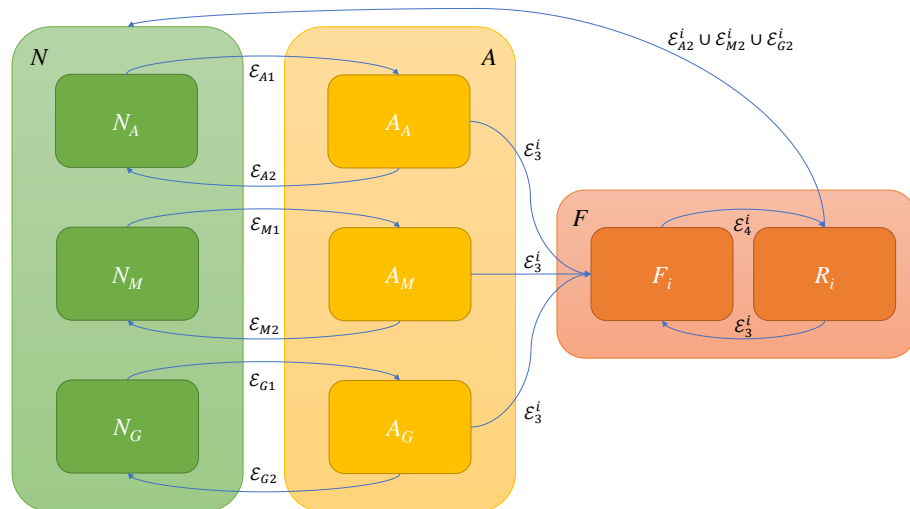


Figure 4. State transition graph for SFDI. Normal state (green) $N = N_A \cap N_M \cap N_G$, alert state (yellow) $A = A_A \cap A_M \cap A_G$, fault state (red) $F = \left(\bigcap_{i=1}^2 F_i\right) \cap \left(\bigcap_{i=1}^2 R_i\right)$. Arrows denote event-driven state transitions.

6. Numerical Simulations Based on Experimental Data

Experimental validation of the proposed approach was carried out with the aim of proving its effectiveness along with its applicability to a low-cost small-UAV architecture. For this purpose, the hardware platform presented in Section 2 was installed on a light tri-rotor aircraft of less than 2 kg (see Figure 5), to gather in-flight sensor data.

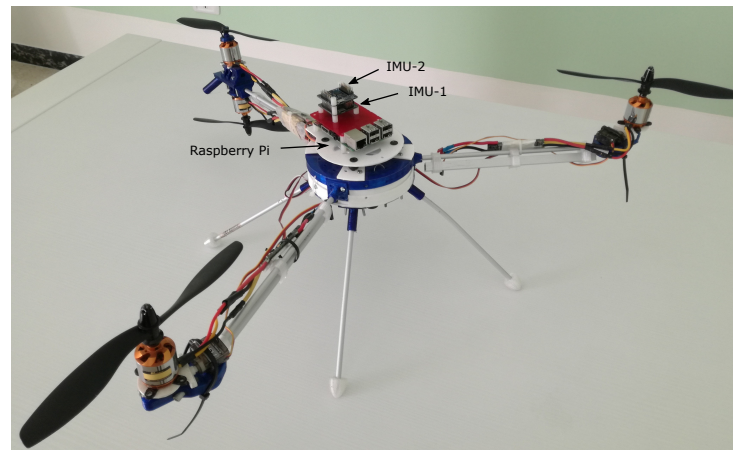


Figure 5. A picture of the custom tri-rotor aircraft used in the flight test with the duplex hardware platform on top.

The tri-rotor platform has an on-board flight controller [79], used to perform an indoor flight, forcing the system to follow pre-assigned reference angles and altitudes.

Captured data from both IMUs is shown in Figure 6. The estimated attitude in the absence of any fault is shown in Figure 6d. Three maneuvers were performed, sequentially changing Euler angles:

- a backward movement, executed by applying a doublet (+/−) to the pitch angle;
- a lateral movement, with a doublet (+/−) applied to the roll angle;
- a doublet applied to the yaw angle.

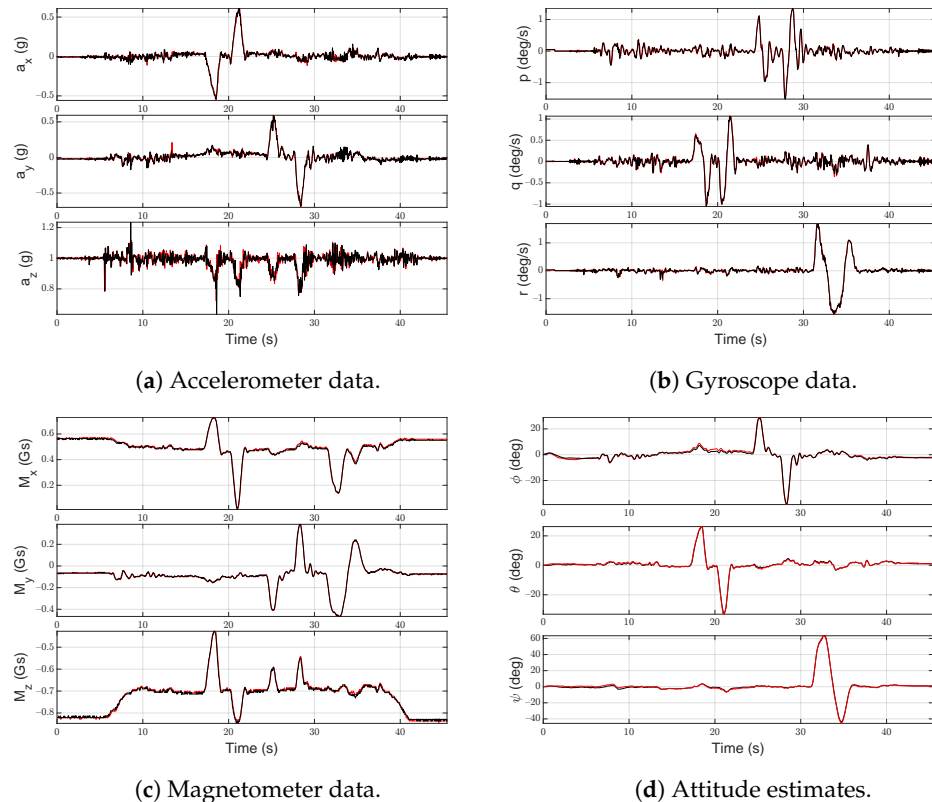


Figure 6. Data gathered from both IMUs during the flight test in the absence of faults. Black lines indicate data gathered by IMU-1 while red lines are for IMU-2.

6.1. Fault Scenario Analyses

Experimental data gathered during the flight were used as measurements in numerical simulations, where realistic faults were injected to test the SFDI algorithm.

Typical fault scenarios [80–84] were considered:

- *F1: intermittent abrupt bias*—a step disturbance of 0.2 g is added to $a_{y,S}^{(1)}(t)$ for $t \in [3, 13]$ s;
- *F2: slow drift*—a linearly increasing signal with a rate of 0.2 rad/s² is added to the measured yaw rate on IMU-2 starting from the time instant $t = 3.5$ s;
- *F3: abrupt freezing*—magnetometer data $M_{x,S}^{(1)}(t)$ stops being updated at $t = 3$ s;
- *F4: oscillation*—a sinusoidal signal with a frequency of 30 rad/s is added to the $r_S^{(1)}(t)$ measurement for $t \in [3, 28]$ s;
- *F5: random walk*—Gaussian noise, whose amplitude is smaller than the relevant detection threshold, is integrated and added to accelerometer data $a_{x,S}(t)$ for $t \in [23, 33]$ s.

Table 2 resumes the values of the SFDI scheme design parameters, as introduced in Sections 4 and 5.

Table 2. Parameters used in numerical results ($\mathcal{I}_{4 \times 4}$ represents the identity matrix of size 4×4).

Parameter	Value
Δ	4096
\mathbf{v}	$[0.0633 \quad 0.3291 \quad 0.5291 \quad 0.2658 \quad 0.2658 \quad 0.2658]^T$
τ_a	0.2411
τ_g	0.2853
τ_m	0.0591
Ω	$24.213 \cdot \mathcal{I}_{4 \times 4}$

6.1.1. Scenario F1—Intermittent Abrupt Bias

The intermittent abrupt bias on $a_{y,S}^{(1)}(t)$ is depicted in Figure 7a. As shown in Figure 7b, the attitude estimated by IMU-1 is affected by an error that mostly affects the roll angle. In Figure 7b, vertical dashed red lines indicate the range in which the fault is injected and the gray area is used to highlight the time interval in which the SFDI algorithm is in the fault state.

As expected, the considered fault is quickly ($t_d = 0.27$ s) detected and IMU-2 is correctly labeled as the operational unit (see red line in Figure 7b). Furthermore, at $t = 13$ s, the accelerometer returns to operational and after 6.96 s the SFDI algorithm returns to the normal state, restoring estimated attitude based on IMU-1 measurements. The time required to restore the estimated attitude based on IMU-1 depends on the design parameters, see Section 6.2. However, it does not affect the performance of the estimated attitude, given the presence of IMU-2.

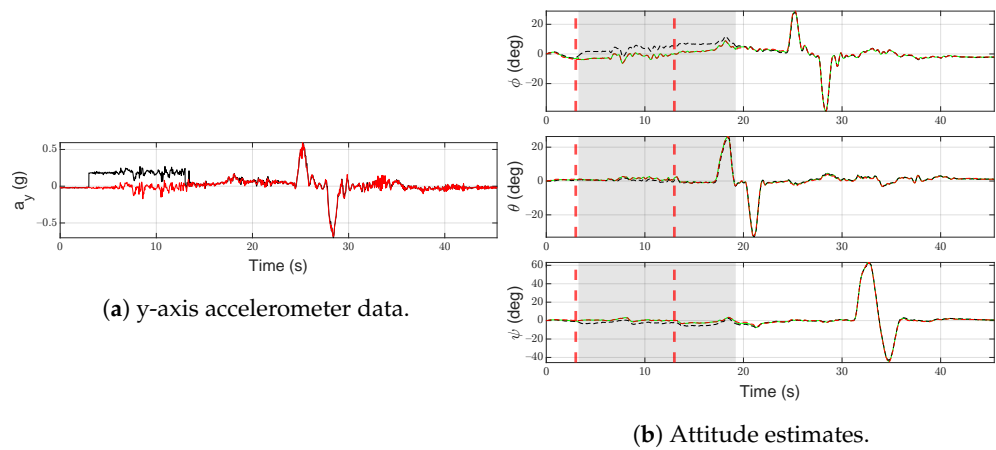


Figure 7. Scenario F1. Black lines denote measurements/estimates from IMU-1, red lines are for measurements/estimates from IMU-2. Green lines are for SFDI output, gray areas highlight the time interval in which the SFDI algorithm is in the fault state. Vertical red dashed lines denote the time instants when the fault occurs and ends.

6.1.2. Scenario F2—Slow Drift

In Figure 8a, the slow drift injected at time $t = 3.5$ s, with a slope of 0.2 rad/s^2 , into the z-axis gyroscope measurement $r_S^{(2)}(t)$ of IMU-2 is shown.

As shown in Figure 8c, at time $t = 4.61$ s, the SFDI scheme isolates the gyroscope fault and moves to the fault state. As IMU-2 does not return to an operational state, the SFDI algorithm persists in the same condition until the end of simulation. A slow drift is difficult to detect and isolate, due to the small slope, causing a delay between the beginning of the fault and the isolation. However, the correction of the magnetometer and the use of the average attitude (green line) in the normal state limit the error in the yaw estimation of some degrees (<3 deg).

Figure 8b shows the norm of the estimated gyroscope biases for both IMUs. It is worth noting that the bias estimated by IMU-2 is increasing over time due to the fault, while IMU-1 is almost hidden by the x-axis.

6.1.3. Scenario F3—Abrupt Freezing

Figure 9a shows x-axis magnetometer data $M_x^{(1)}$ from IMU-1 subject to an abrupt freezing over almost the whole simulation.

Results in Figure 9b show that the SFDI algorithm switches between normal and fault states several times, when the broken value of $M_x^{(1)}$ is similar to the correct one and estimated attitudes are in the predefined threshold. Furthermore, it is worth noting that the fault was isolated with a significant delay. This behavior depends on magnetometer signals $M_x^{(1)}$ and $M_x^{(2)}$ that, in the first phases of the flight, are almost the same. At the take-off, $M_x^{(2)}$ is affected by a small change due to the magnetic field of the electric motors that is altered by an increasing power request. Because the faulted IMU is frozen, only at the take-off is the SFDI algorithm able to detect the fault. After that, at several times during the flight, both errors on attitudes and magnetometer signals are below the predefined threshold, causing the return to the normal state. However, the reliability of the overall system is not compromised because, as shown in Figure 9b, a sensor fault does not affect attitude estimation for a relevant time interval.

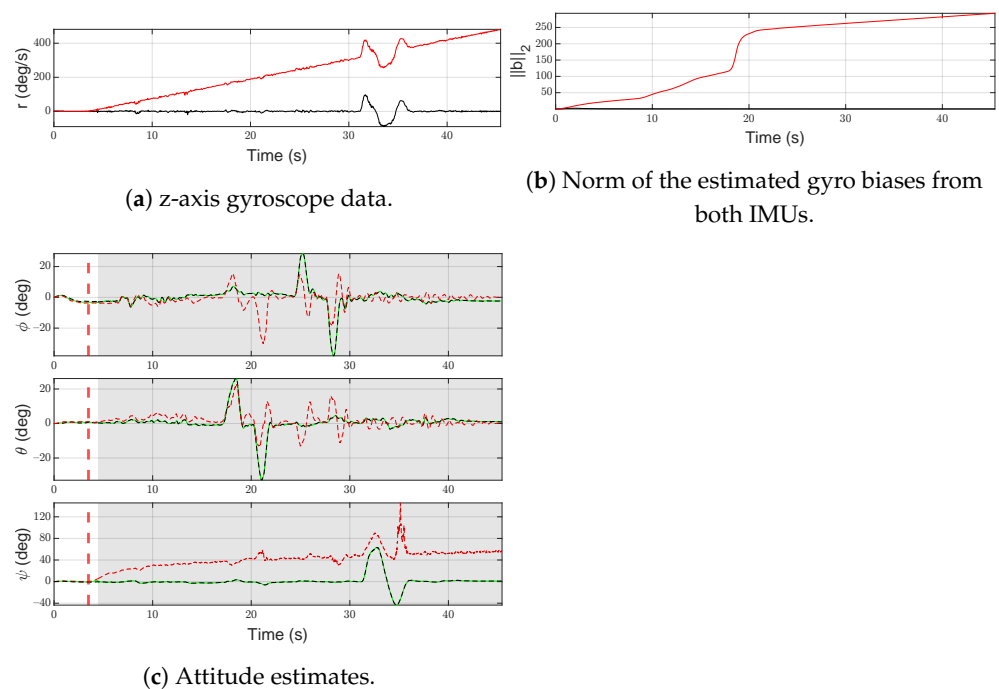


Figure 8. Scenario F2. Black lines denote measurements/estimates from IMU-1, red lines are for measurements/estimates from IMU-2. Green lines represent the output of the system, gray areas highlight the time interval in which the SFDI algorithm is in the fault state. Vertical red dashed lines denote the time instants when the fault occurs and ends.

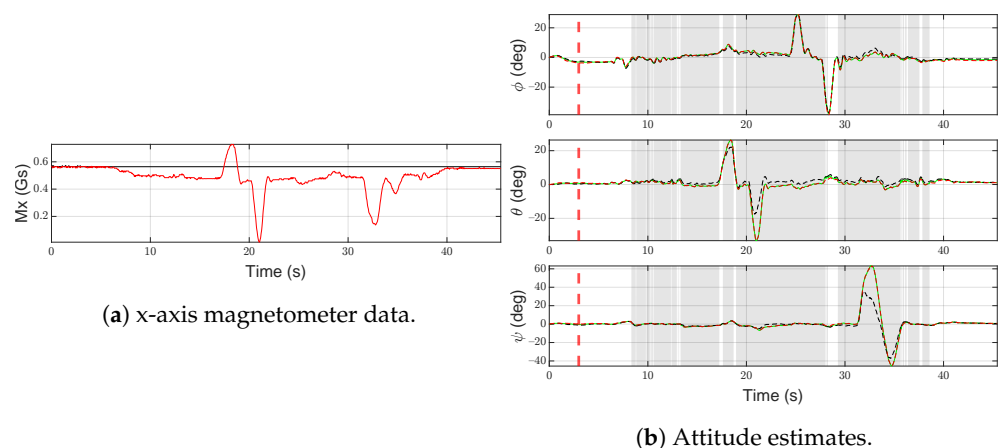


Figure 9. Scenario F3. Black lines denote measurements/estimates from IMU-1, red lines are for measurements/estimates from IMU-2. Green lines represent the output of the system, gray areas highlight the time interval in which the SFDI algorithm is in the fault state. Vertical red dashed lines denote the time instants when the fault occurs and ends.

6.1.4. Scenario F4—Oscillation

In Figure 10a, the considered fault on the z-axis gyroscope measurement $r_S^{(1)}$ is shown. In particular, a sinusoidal disturbance, with a frequency (~ 10 Hz), compliant to tri-rotor dynamics, is added to $r_S^{(1)}$ in the time interval $t \in [3, 28]$ s.

As shown in Figure 10c, although the attitude estimated by IMU-1 is only partially affected by the fault, the SFDI algorithm is able to detect the problem after a very short time, with the transition to the fault state from $t = 3.02$ s to $t = 28.01$ s denoting a fast return to the normal state, once the signal on IMU-1 becomes operational.

In Figure 10b, the norm of the estimated gyro biases is shown, highlighting the difference between IMUs.

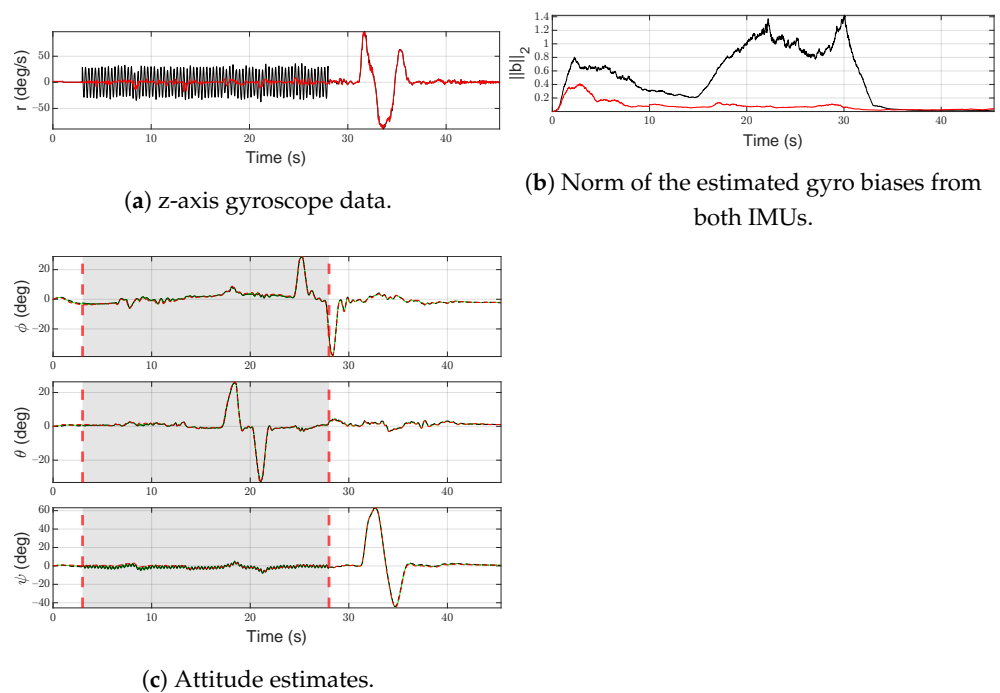


Figure 10. Scenario F4. Black lines denote measurements/estimates from IMU-1, red lines are for measurements/estimates from IMU-2. Green lines represent the output of the system, gray areas highlight the time interval in which the SFDI algorithm is in the fault state. Vertical red dashed lines denote the time instants when the fault occurs and ends.

6.1.5. Scenario F5—Random Walk

In Figure 11a, the effects of the random walk disturbance on accelerometer data are shown. This signal is created by integrating a Gaussian noise with a mean value of 0.07 g and overlapped to the real signal in the time interval [23 s, 33 s]. In order to stress the algorithm, the mean value of noise is lower than the relevant detection threshold.

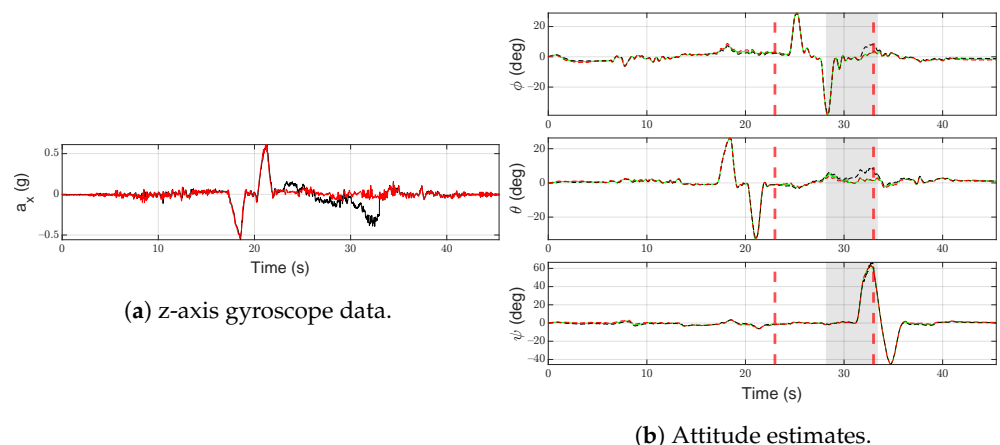


Figure 11. Scenario F5. Black lines denote measurements/estimates from IMU-1, red lines are for measurements/estimates from IMU-2. Green lines represent the output of the system, gray areas highlight the time interval in which the SFDI algorithm is in the fault state. Vertical red dashed lines denote the time instants when the fault occurs and ends.

As shown in Figure 11b, the SFDI algorithm moves to the fault state at $t = 28.16$ s, with a notable delay from the beginning of the injected fault. The reliability of the estimated attitude is not affected and such delay depends on the tri-rotor attitude: without any relevant maneuver affecting a change in the pitch angle, a small error on $a_{x,S}^{(1)}(t)$ results

in a small trace on the attitude (<1 deg in the simulation). After IMU-1 becomes newly operational, the SFDI algorithm returns to the normal state (at $t = 33.41$ s).

6.2. Algorithm Parameter Sensitivity Analysis

Effects of parameters on the performance of the proposed algorithm are discussed by means of a comparison between several choices. Scenario F1 is considered. We denote with $\mathcal{T}_0 = \{300, 301, \dots, 1300\}$ the discrete time set in which the fault is injected. Executing the SFDI algorithm the fault is detected and isolated. With $\mathcal{T}_f = \{k : F_1(k) = 1\}$ being the set of time instants in which IMU-1 is considered faulted, the following performance indices were considered:

1. *computation time*—the time needed to execute the SFDI algorithm at each time instant on the embedded platform Raspberry Pi 3B;
2. *correct detection*—the ratio between the correctly detected fault time interval and the real fault duration, computed as $CD = \text{card}(\mathcal{T}_f \cap \mathcal{T}_0) / \text{card}(\mathcal{T}_0)$, where $\text{card}(\mathcal{T})$ denotes the cardinality of set \mathcal{T} ;
3. *wrong detection*—the ratio between the badly detected fault time interval and the real fault duration, computed as $WD = \text{card}(\mathcal{T}_f \cap \overline{\mathcal{T}}_0) / \text{card}(\mathcal{T}_0)$, where $\overline{\mathcal{T}}_0$ is the complementary set of \mathcal{T}_0 ;
4. *detection time*—the delay between the beginning of the fault and the detection, computed as $DT = (\min \mathcal{T}_f - \min \mathcal{T}_0)$;
5. *recovery time*—the delay between the end of the fault and the fault recovery, computed as $RT = (\max \mathcal{T}_f - \max \mathcal{T}_0)$.

In Table 3, the computing time is shown considering several numbers of particles Δ . Results are taken from 30 executions of the algorithm for each Δ value on the above mentioned Raspberry Pi 3B platform.

Table 3. Raspberry Pi computing time for several numbers of particles (Δ).

Number of Particles	Computation Time (s)	
	Average	Standard Deviation
1024	0.0138	0.0012
2048	0.0303	0.0022
4096	0.0682	0.0030
8192	0.1573	0.0050
16,384	0.3401	0.0071

In Table 4, the effects in terms of variation of 2–5 performance indices for varying numbers of particles are shown. A growth in the number of particles decreases PF convergence time thus leading to a quicker recovery from a fault. However, if the output of both (operational and faulty) IMUs is close, an excessively short convergence speed may cause the mislabeling of a faulty IMU as operational, reducing the *correct detection* ratio. For these reasons increasing the Δ value behind a certain threshold brings no advantages, while increasing computational effort, see Table 3. In view of the sensitivity analysis results, the chosen value of Δ is boldfaced. It represents the number of particles, among the considered values, showing the best correct detection ratio while guaranteeing real-time execution.

Table 4. Sensitivity analysis on Δ .

Number of Particles	Correct Detection (%)	Wrong Detection (%)	Detection Time (s)	Recovery Time (s)
1024	89.81	0.00	0.27	11.56
4096	99.50	0.00	0.27	6.96
16,384	92.81	0.30	0.27	4.07
32,768	92.81	0.00	0.27	4.04

In Table 5 the effects of τ_a are evaluated. As expected, reducing the fault detection threshold may increase the number of wrong detections, whereas high threshold values may increase the number of missed detections. Detection time appears to be zero with lower τ_a values because of false positives and the fact that recovery time may be affected by false detections. The chosen value of τ_a is boldfaced and it represents the trade-off, among the considered thresholds, which shows the best ratio between the *correct detection* index and the *wrong identification* one.

Table 5. Sensitivity analysis for varying of τ_a .

$\alpha \cdot \tau_a$	Correct Detection (%)	Wrong Detection (%)	Detection Time (s)	Recovery Time (s)
$\alpha = 0.25$	99.40	24.36	0.00	18.67
$\alpha = 0.75$	99.90	0.06	0.00	6.96
$\alpha = 1$	99.50	0.00	0.27	6.96
$\alpha = 1.5$	45.75	1.03	0.27	6.96

In Table 6 the influence of Ω is shown. A higher recovery threshold value leads to a shorter recovery time, which eventually leads to exiting the fault condition too early; it causes an increase in the number of missed alarms, reducing SFDI algorithm sensitivity. The chosen value of Ω is boldfaced and it represents the best threshold value in terms of acceptable recovery time.

Table 6. Sensitivity analysis for varying of Ω .

$\alpha \cdot \Omega$	Correct Detection (%)	Wrong Detection (%)	Detection Time (s)	Recovery Time (s)
$\alpha = 0.25$	97.30	0.00	0.27	10.28
$\alpha = 0.5$	97.30	0.00	0.27	7.61
$\alpha = 0.75$	97.30	0.00	0.27	7.44
$\alpha = 1$	99.50	0.00	0.27	6.96
$\alpha = 4$	35.26	0.84	0.27	0.00

Finally, in Table 7 the effect of the v parameter is evaluated. It affects PF convergence time in two opposite ways. Larger v values may boost the convergence speed by enlarging the region where the algorithm searches for the best state estimate, but it can also reduce the correct match ratio in case of faults whose magnitude is close to the relevant detection threshold. Neither the wrong identification ratio nor the detection time are influenced by v value. The chosen value of v is boldfaced and it represents the trade-off in terms of the ratio between the *correct detection* and the *wrong identification* indices.

Table 7. Sensitivity analysis for varying of v .

$\alpha \cdot v$	Correct Detection (%)	Wrong Detection (%)	Detection Time (s)	Recovery Time(s)
$\alpha = 0.25$	88.11	0.00	0.27	32.45
$\alpha = 0.5$	85.71	0.00	0.27	4.43
$\alpha = 0.75$	43.36	0.00	0.27	0.00
$\alpha = 1$	99.50	0.00	0.27	6.96
$\alpha = 4$	94.31	0.00	0.27	4.08

7. Conclusions

In commercial unmanned aviation, an increase in the system reliability is needed in order to make flight risks acceptable.

With this aim, in this paper, the implementation of a sensor fault detection and isolation technique on low-cost hardware typical of commercial micro- and mini-UAVs was performed. Given the crucial role of attitude estimation in flying platform stabilization, the proposed procedure is aimed at detecting and isolating faults on inertial measurement

units used to estimate UAV attitude. The hardware is based on a duplex architecture with two inertial measurement units (IMUs) including 3-axis gyroscopes, accelerometers and magnetometers. A particle filtering approach is proposed to deal with the attitude estimation problem, being able to deal with non-linear motion models and non-Gaussian noise distributions.

Results show the effectiveness of such an approach. Data gathered from a real flight of a tri-rotor flying platform were useful to test the SFDI algorithm with real sensor disturbances due to vibrations and changes to magnetic field. The proposed algorithm correctly detected every injected fault, including for challenging scenarios such as freezing faults and slow drifts.

The proposed architecture increases system reliability by doubling the number of on-board sensors. This produces a reduction in weight, costs and power consumption, in comparison to the gold standard of triplex architecture for avionics at the price of lower speed and accuracy.

In the literature, few duplex IMU architectures have been proposed. In Table 8, the authors compared the proposed SFDI algorithm with the SFDI logic presented in a previous work [72], to highlight the pros and cons of the current solution by comparing platforms in terms of detection time (t_d) and maximum attitude error (e_{max}).

Table 8. Comparison between duplex architectures.

X	PF + SFDI		Previous Work [72]	
	t_d (s)	e_{max} (rad)	t_d (s)	e_{max} (rad)
F1	0.270	0.018 (on ϕ angle)	0.094	0.073 (on ϕ angle)
F2	0.960	0.024 (on ψ angle)	2.075	0.026 (on ψ angle)

The implementation of the algorithm on a Raspberry Pi low-cost embedded platform with two IMU boards has shown that, although a particle filter is considered a heavy algorithm, the proposed procedure can be used on low-cost unmanned aircraft, with an increase in system reliability and without a significant change in the overall cost.

To make the algorithm readily available and easily tunable, a sensitivity analysis of algorithm parameters on the SFDI performance was conducted, highlighting the chosen settings.

For the latest low-cost drones equipped with sonar, lidar and optical flow sensors, the future applications of our procedure are currently under research, using more sensors and also validating duplex heterogeneous architectures, where redundant sensor platforms can contain different sensors.

Author Contributions: Conceptualization, E.D., V.A.N., I.N. and V.S.; data curation, E.D., V.A.N., I.N. and V.S.; formal analysis, E.D., V.A.N., I.N. and V.S.; investigation, E.D., V.A.N., I.N. and V.S.; methodology, E.D., V.A.N., I.N. and V.S.; resources, E.D., V.A.N., I.N. and V.S.; software, E.D., V.A.N., I.N. and V.S.; validation, E.D., V.A.N., I.N. and V.S.; visualization, E.D., V.A.N., I.N. and V.S.; writing—original draft, E.D., V.A.N., I.N. and V.S.; writing—review and editing, E.D., V.A.N., I.N. and V.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Adams, S.M.; Friedland, C.J. A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management. In Proceedings of the 9th International Workshop on Remote Sensing for Disaster Response, Stanford, CA, USA, 15–16 September 2011; Volume 8.
2. Danilov, A.; Smirnov, U.D.; Pashkevich, M. The system of the ecological monitoring of environment which is based on the usage of UAV. *Russ. J. Ecol.* **2015**, *46*, 14–19. [[CrossRef](#)]
3. Furci, M.; Paoli, A.; Naldi, R. A supervisory control strategy for robot-assisted search and rescue in hostile environments. In Proceedings of the 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), Cagliari, Italy, 10–13 September 2013; pp. 1–4.
4. Shakhathreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. *IEEE Access* **2019**, *7*, 48572–48634. [[CrossRef](#)]
5. Peng, Z.; Kato, T.; Takahashi, H.; Kinoshita, T. Intelligent home security system using agent-based IoT devices. In Proceedings of the 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 27–30 October 2015; pp. 313–314.
6. Marchese, M.; Moheddine, A.; Patrone, F. IoT and UAV integration in 5G hybrid terrestrial-satellite networks. *Sensors* **2019**, *19*, 3704. [[CrossRef](#)] [[PubMed](#)]
7. Schaefer, R. *Unmanned Aerial Vehicle Reliability Study*; Office of the Secretary of Defense: Washington, DC, USA, 2003.
8. Hwang, I.; Kim, S.; Kim, Y.; Seah, C.E. A survey of fault detection, isolation, and reconfiguration methods. *IEEE Trans. Control Syst. Technol.* **2010**, *18*, 636–653. [[CrossRef](#)]
9. Marzat, J.; Piet-Lahanier, H.; Damongeot, F.; Walter, E. Model-based fault diagnosis for aerospace systems: A survey. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2012**, *226*, 1329–1360. [[CrossRef](#)]
10. Zhang, Y.; Jiang, J. Bibliographical review on reconfigurable fault-tolerant control systems. *Annu. Rev. Control* **2008**, *32*, 229–252. [[CrossRef](#)]
11. Fekih, A. Fault Diagnosis and Fault Tolerant Control Design for aerospace Systems: A Bibliographical Review. In Proceedings of the American Control Conference (ACC), Portland, OR, USA, 4–6 June 2014; Volume 32.
12. Chen, H.; Wang, X.; Li, Y. A Survey of Autonomous Control for UAV. In Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence, Shanghai, China, 7–8 November 2009; Volume 2, pp. 267–271.
13. Zhang, Y.; Jiang, J. Issues on Integration of Fault Diagnosis and Reconfigurable Control in Active Fault-Tolerant Control Systems. *IFAC Proc. Vol.* **2006**, *39*, 1437–1448. [[CrossRef](#)]
14. Hansen, S.; Blanke, M.; Adrian, J. Fault Diagnosis and Fault Handling for Autonomous Aircraft. Ph.D. Thesis, Department of Electrical Engineering, Hong Kong, China, 2012.
15. Schulte, P.Z.; Spencer, D.A. State Machine Fault Protection Architecture for Aerospace Vehicle Guidance, Navigation, and Control. *J. Aerosp. Inf. Syst.* **2020**, *17*, 70–85. [[CrossRef](#)]
16. Boskovic, J.; Jackson, J.A. An Innovative Approach to Air Data Sensor FDIR for Commercial Aircraft. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, San Diego, CA, USA, 4–8 January 2016; p. 0375.
17. Blanke, M.; Hansen, S. Towards self-tuning residual generators for UAV control surface fault diagnosis. In Proceedings of the 2013 Conference on Control and Fault-Tolerant Systems (SysTol), Nice, France, 9–11 October 2013; pp. 37–42.
18. Suarez, A.; Heredia, G.; Ollero, A. Cooperative Virtual Sensor for Fault Detection and Identification in Multi-UAV Applications. *J. Sensors* **2018**, *2018*, 1–19. [[CrossRef](#)]
19. Leung, T.; Rife, J.; Seiler, P.; Venkataraman, R. Comparison of fault-tree models for fault detection, isolation, and recovery algorithms. *J. Aerosp. Inf. Syst.* **2017**, *14*, 517–522. [[CrossRef](#)]
20. Crassidis, J.; Markley, L.; Cheng, Y. Survey of Nonlinear Attitude Estimation Methods. *J. Guid. Control Dyn.* **2007**, *30*, 12–28. [[CrossRef](#)]
21. Gośliński, J.; Giernacki, W.; Królikowski, A. A Nonlinear Filter for Efficient Attitude Estimation of Unmanned Aerial Vehicle (UAV). *J. Intell. Robot. Syst.* **2019**, *95*, 1079–1095. [[CrossRef](#)]
22. Zhong, Y.; Zhang, W.; Zhang, Y.; Zuo, J.; Zhan, H. Sensor Fault Detection and Diagnosis for an Unmanned Quadrotor Helicopter. *J. Intell. Robot. Syst.* **2019**, *96*, 555–572. [[CrossRef](#)]
23. Goupil, P. AIRBUS state of the art and practices on FDI and FTC in flight control system. *Control Eng. Pract.* **2011**, *19*, 524–539. [[CrossRef](#)]
24. Rudin, K.; Ducard, G.J.; Siegwart, R.Y. A sensor fault detection for aircraft using a single Kalman filter and hidden Markov models. In Proceedings of the 2014 IEEE Conference on Control Applications (CCA), Juan Les Antibes, France, 8–10 October 2014; pp. 991–996.
25. Berdjag, D.; Cieslak, J.; Zolghadri, A. Fault diagnosis and monitoring of oscillatory failure case in aircraft inertial system. *Control Eng. Pract.* **2012**, *20*, 1410–1425. [[CrossRef](#)]
26. Rosa, P.; Silvestre, C. Fault detection and isolation of LPV systems using set-valued observers: An application to a fixed-wing aircraft. *Control Eng. Pract.* **2013**, *21*, 242–252. [[CrossRef](#)]
27. Notaro, I.; Ariola, M.; D’Amato, E.; Mattei, M. HW VS SW sensor redundancy: Fault detection and isolation observer based approaches for inertial measurement units. In Proceedings of the 29th Congress of International Council of Aeronautical Sciences, St. Peterburg, Russia, 7–12 September 2014.

28. Ariola, M.; Corraro, F.; Mattei, M.; Notaro, I.; Sollazzo, A. An SFDI observer-based scheme for a general aviation aircraft. In Proceedings of the 2013 Conference on Control and Fault-Tolerant Systems (SysTol), Nice, France, 9–11 October 2013; pp. 152–157.
29. Alwi, H.; Edwards, C. Robust sensor fault estimation for tolerant control of a civil aircraft using sliding modes. In Proceedings of the American Control Conference, Minneapolis, MN, USA, 14–16 June 2006; pp. 5704–5709.
30. Varga, A.; Ossmann, D.; Joos, H. A fault diagnosis based reconfigurable longitudinal control system for managing loss of air data sensors for a civil aircraft. In Proceedings of the 19th IFAC World Congress 2014, Cape Town, South Africa, 24–29 August 2014; Volume 47, pp. 3489–3496.
31. Mattei, M.; Ollio, L.; Scordamaglia, V. A set based approach for SFDI on small commercial aircraft. In Proceedings of the 2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol), Barcelona, Spain, 7–9 September 2016; pp. 678–683.
32. Mattei, M.; Paviglianiti, G.; Scordamaglia, V. Nonlinear observers with H_∞ performance for sensor fault detection and isolation: A linear matrix inequality design procedure. *Control Eng. Pract.* **2005**, *13*, 1271–1281. [\[CrossRef\]](#)
33. Mattei, M.; Paviglianiti, G.; Scordamaglia, V. Nonlinear H ∞ identity observers for sensor FDI: An application example. In Proceedings of the ICM'04—IEEE International Conference on Mechatronics, Istanbul, Turkey, 5 June 2004.
34. Doucet, A.; Godsill, S.; Andrieu, C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comput.* **2000**, *10*, 197–208. [\[CrossRef\]](#)
35. Cheng, Y.; Crassidis, J. Particle Filtering for Sequential Spacecraft Attitude Estimation. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, RI, USA, 16–19 August 2004.
36. Oshman, Y.; Carmi, A. Attitude Estimation from Vector Observations Using a Genetic-Algorithm-Embedded Quaternion Particle Filter. *J. Guid. Control Dyn.* **2006**, *29*, 879–891. [\[CrossRef\]](#)
37. Carmi, A.; Oshman, Y. Adaptive Particle Filtering for Spacecraft Attitude Estimation from Vector Observations. *J. Guid. Control Dyn.* **2009**, *32*, 232–241. [\[CrossRef\]](#)
38. Jiang, X.-Y.; Ma, G.-F. Spacecraft attitude estimation from vector measurements using particle filter. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; Volume 2, pp. 682–687.
39. Chagas, R.A.J.; Waldmann, J. Rao-Blackwellized particle filter with vector observations for satellite three-axis attitude estimation and control in a simulated testbed. *Sba: Controle Automaç Ao Soc. Bras. De Autom.* **2012**, *23*, 277–293. [\[CrossRef\]](#)
40. Chagas, R.; Waldmann, J. Nonlinear filtering in a simulated three-axis satellite attitude estimation and control tested. *J. Aerosp. Eng. Sci. Appl.* **2010**, *2*, 37–49. [\[CrossRef\]](#)
41. Kadirkamanathan, V.; Li, P.; Jaward, M.H.; Fabri, S.G. A sequential Monte Carlo filtering approach to fault detection and isolation in nonlinear systems. In Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187), Sydney, NSW, Australia, 12–15 December 2000; Volume 5, pp. 4341–4346.
42. Kadirkamanathan, V.; Li, P.; Jaward, M.H.; Fabri, S.G. Particle filtering-based fault detection in non-linear stochastic systems. *Int. J. Syst. Sci.* **2002**, *33*, 259–265. [\[CrossRef\]](#)
43. Dearden, R.; Clancy, D. *Particle Filters for Real-Time Fault Detection in Planetary Rovers*; Technical Report; National Aeronautics and Space Administration: Washington, DC, USA, 2002.
44. Wang, E.; Jia, C.; Tong, G.; Qu, P.; Lan, X.; Pang, T. Fault detection and isolation in GPS receiver autonomous integrity monitoring based on chaos particle swarm optimization-particle filter algorithm. *Adv. Space Res.* **2018**, *61*, 1260–1272. [\[CrossRef\]](#)
45. Wei, T.; Huang, Y.; Chen, C.L.P. Adaptive Sensor Fault Detection and Identification Using Particle Filter Algorithms. *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.* **2009**, *39*, 201–213. [\[CrossRef\]](#)
46. Duan, Z.; Cai, Z.; Yu, J. Adaptive Particle Filter for Unknown Fault Detection of Wheeled Mobile Robots. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 1312–1315.
47. Yu, J. A particle filter driven dynamic Gaussian mixture model approach for complex process monitoring and fault diagnosis. *J. Process Control* **2012**, *22*, 778–788. [\[CrossRef\]](#)
48. Zhang, Q.; Campillo, F.; Cerou, F.; Legland, F. Nonlinear system fault detection and isolation based on bootstrap particle filters. In Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 15 December 2005; pp. 3821–3826.
49. Yin, S.; Zhu, X. Intelligent Particle Filter and Its Application to Fault Detection of Nonlinear System. *IEEE Trans. Ind. Electron.* **2015**, *62*, 3852–3861. [\[CrossRef\]](#)
50. Hamzah, N.H.; Yaacob, S.; Muthusamy, H.; Hamzah, N. Comparative Study of Extended Kalman Filter and Particle Filter for Attitude Estimation in Gyroless Low Earth Orbit Spacecraft. In *Advanced Computational Methods for Knowledge Engineering*; Le Thi, H.A., Nguyen, N.T., Do, T.V., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 95–106.
51. Zhang, C.; Taghvaei, A.; Mehta, P.G. Attitude estimation with feedback particle filter. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 5440–5445.
52. Hernández, F.; Giribet, J.I. Comparison between linearized Kalman filters and particle filters applied to integrated navigation systems. In Proceedings of the 6th International ESA Conference on Guidance, Navigation and Control Systems, Loutraki, Greece, 17–20 October 2005.
53. Ahmed, H.; Tahir, M. Accurate Attitude Estimation of a Moving Land Vehicle Using Low-Cost MEMS IMU Sensors. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1723–1739. [\[CrossRef\]](#)
54. Hua, M.D.; Ducard, G.; Hamel, T.; Mahony, R. Introduction to Nonlinear Attitude Estimation for Aerial Robotic Systems. *Aerosp. Lab* **2014**, AL08-04.

55. D'Amato, E.; Notaro, I.; Mattei, M.; Tartaglione, G. Attitude and position estimation for an UAV swarm using consensus Kalman filtering. In Proceedings of the 2015 IEEE Metrology for Aerospace (MetroAeroSpace), Benevento, Italy, 4–5 June 2015; pp. 519–524.
56. Hyyti, H.; Visala, A. A DCM Based Attitude Estimation Algorithm for Low-Cost MEMS IMUs. *Int. J. Navig. Obs.* **2015**, *2015*, 503814. [[CrossRef](#)]
57. Costanzi, R.; Fanelli, F.; Monni, N.; Ridolfi, A.; Allotta, B. An attitude estimation algorithm for mobile robots under unknown magnetic disturbances. *IEEE/ASME Trans. Mechatronics* **2016**, *21*, 1900–1911. [[CrossRef](#)]
58. De Marina, H.G.; Espinosa, F.; Santos, C. Adaptive UAV attitude estimation employing unscented Kalman filter, FOAM and low-cost MEMS sensors. *Sensors* **2012**, *12*, 9566–9585. [[CrossRef](#)]
59. Chiella, A.C.; Teixeira, B.O.; Pereira, G.A. Quaternion-based robust attitude estimation using an adaptive unscented Kalman filter. *Sensors* **2019**, *19*, 2372. [[CrossRef](#)]
60. Cheng, Y.; Crassidis, J.L. Particle filtering for attitude estimation using a minimal local-error representation. *J. Guid. Control. Dyn.* **2010**, *33*, 1305–1310. [[CrossRef](#)]
61. Bar-Itzhack, I.Y. REQUEST-A recursive QUEST algorithm for sequential attitude determination. *J. Guid. Control. Dyn.* **1996**, *19*, 1034–1038. [[CrossRef](#)]
62. Psiaki, M.L. Attitude-determination filtering via extended quaternion estimation. *J. Guid. Control. Dyn.* **2000**, *23*, 206–214. [[CrossRef](#)]
63. Euston, M.; Coote, P.; Mahony, R.; Kim, J.; Hamel, T. A complementary filter for attitude estimation of a fixed-wing UAV. In Proceedings of the 2008 IEEE/RSJ international conference on intelligent robots and systems, Nice, France, 22–26 September 2008; pp. 340–345.
64. Wu, J.; Zhou, Z.; Fourati, H.; Li, R.; Liu, M. Generalized linear quaternion complementary filter for attitude estimation from multisensor observations: An optimization approach. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1330–1343. [[CrossRef](#)]
65. Madgwick, S. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Rep. X-Io Univ. Bristol (UK)* **2010**, *25*, 113–118.
66. Blanco, J.L. A tutorial on se (3) transformation parameterizations and on-manifold optimization. *Univ. Malaga, Tech. Rep* **2010**, *3*, 6.
67. Valenti, R.G.; Dryanovski, I.; Xiao, J. Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs. *Sensors* **2015**, *15*, 19302–19330. [[CrossRef](#)]
68. Park, S.; Horowitz, R.; Tan, C.W. Dynamics and control of a MEMS angle measuring gyroscope. *Sensors Actuators A Phys.* **2008**, *144*, 56–63. [[CrossRef](#)]
69. Vitali, A. *Noise Analysis and Identification in MEMS Sensors, Allan, Time, Hadamard, Overlapping, Modified, Total Variance*; Technical Report DT0064; STMicroelectronics: Geneva, Switzerland, 2016.
70. Hoffmann, F.; Goddemeier, N.; Bertram, T. Attitude estimation and control of a quadcopter. In Proceedings of the 2010 IEEE/RSJ International Conf. on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010.
71. D'Amato, E.; Mattei, M.; Notaro, I.; Scordamaglia, V. UAV Sensor FDI in Duplex Attitude Estimation Architectures Using a Set-Based Approach. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 2465–2475. [[CrossRef](#)]
72. D'Amato, E.; Mattei, M.; Mele, A.; Notaro, I.; Scordamaglia, V. Fault tolerant low cost IMUS for UAVs. In Proceedings of the 2017 IEEE International Workshop on Measurement and Networking (MN), Naples, Italy, 27–29 September 2017; pp. 1–6.
73. Simon, D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
74. Gordon, N.J.; Salmond, D.J.; Smith, A.F. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proc. Radar Signal Process.* **1993**, *140*, 107–113. [[CrossRef](#)]
75. Georgy, J.; Noureldin, A.; Korenberg, M.; Bayoumi, M. Modeling the Stochastic Drift of a MEMS-Based Gyroscope in Gyro/Odometer/GPS Integrated Navigation. *Intell. Transp. Syst. IEEE Trans.* **2011**, *11*, 856–872. [[CrossRef](#)]
76. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005.
77. Dellaert, F.; Fox, D.; Burgard, W.; Thrun, S. Monte Carlo localization for mobile robots. In Proceedings of the 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Detroit, MI, USA, 10–15 May 1999; Volume 2, pp. 1322–1328.
78. Li, T.; Sun, S.; Sattar, T.P.; Corchado, J.M. Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Syst. Appl.* **2014**, *41*, 3944–3954. [[CrossRef](#)]
79. D'Amato, E.; Di Francesco, G.; Notaro, I.; Tartaglione, G.; Mattei, M. Nonlinear Dynamic Inversion and Neural Networks for a Tilt Tri-Rotor UAV. *IFAC-PapersOnLine* **2015**, *48*, 162–167. [[CrossRef](#)]
80. Rogne, R.H.; Johansen, T.A.; Fossen, T.I. Observer and IMU-based detection and isolation of faults in position reference systems and gyrocompasses with dual redundancy in dynamic positioning. In Proceedings of the 2014 IEEE Conference on Control Applications (CCA), Juan Les Antibes, France, 8–10 October 2014; pp. 83–88.
81. Lu, P.; Van Eykeren, L.; van Kampen, E.; de Visser, C.C.; Chu, Q.P. Aircraft Inertial Measurement Unit Fault Identification with Application to Real Flight Data. *J. Guid. Control. Dyn.* **2015**, *38*, 2467–2475. [[CrossRef](#)]
82. Kolcio, K.O. Model-Based Fault Detection and Isolation System for Increased Autonomy. In Proceedings of the AIAA SPACE 2016, AIAA SPACE, Long Beach, CA, USA, 13–16 September 2016.

-
83. Furlas, G.K.; Karras, G.C.; Kyriakopoulos, K.J. Sensors fault diagnosis in autonomous mobile robots using observer—Based technique. In Proceedings of the 2015 International Conference on Control, Automation and Robotics, Singapore, 20–22 May 2015; pp. 49–54.
 84. Bhatti, U.I.; Ochieng, W.Y. Failure Modes and Models for Integrated GPS/INS Systems. *J. Navig.* **2007**, *60*, 327–348. [[CrossRef](#)]