Received: 5 January 2021 | Revised: 13 June 2021 | Accepted: 5 July 2021

DOI: 10.1049/cth2.12180

IET Control Theory & Applications

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Flight evaluation of simultaneous actuator/sensor fault reconstruction on a quadrotor minidrone

S. Waitman[1] | **H. Alwi**[2] | **C. Edwards**[2]

[1] ONERA - The French Aerospace Lab, DTIS, Toulouse, France

[2] College of Engineering, Mathematics and Physical Sciences, University of Exeter, Devon, Exeter, UK

Correspondence
H. Alwi, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter EX4 4QF, Devon, UK.
Email: h.alwi@exeter.ac.uk

This work is part of the CRUISE project (Grant EP/P015409/1) supported by the UK Engineering and Physical Sciences Research Council (EPSRC)

Funding information
Engineering and Physical Sciences Research Council, Grant/Award Number: EP/P015409/1

Abstract

In view of the increase in the number of Unmanned Aerial Vehicles (UAVs) in the commercial and private sectors, it is imperative to make sure that such systems are safe, and thus resilient to faults and failures. This paper considers the numerical design and practical implementation of a linear parameter-varying (LPV) sliding mode observer for Fault Detection and Diagnosis (FDD) of a quadrotor minidrone. Starting from a nonlinear model of the minidrone, an LPV model is extracted for design, and the observer synthesis procedure, using Linear Matrix Inequalities (LMI), is detailed. Simulations of the observer FDD show good performance. The observer is then implemented on a Parrot® Rolling Spider minidrone and a series of flight tests is performed to assess the FDD capabilities in real time using its on-board processing power. The flight tests confirm the performance obtained in simulation, and show that the sliding mode observer is able to provide reliable fault reconstruction for quadrotor minidrone systems.

1 | INTRODUCTION

Unmanned Aerial Vehicles (UAVs) now permeate the commercial and private sectors, and their low cost and ease of deployment indicate there is great potential for current and future applications. The estimated value of the UAV market is expected to grow by more than 30% from 2016 to 2020 [1]. Multicopters are expected to play an important part in these figures, particularly in the commercial and private sectors, with uses ranging from aerial filming and photography to agriculture, delivery and construction, to name but a few [1]. As their uses increase, so does the fleet size, and agencies around the world are pressing for regulations to improve privacy and, above all, security. In contrast to what is found in civil aviation, where the human factor is preponderant, analysis of incidents involving UAVs suggest that around 62% of safety issues originate in equipment problems, with the main occurrences being operational damage and loss of control in flight [2]. This data shows the importance of developing technology that allows UAVs (and/or their pilots) to cope with faults and failure during flight.

Resilience against such events can be sought by means of Fault-Tolerant Control (FTC) techniques. This area of research continues to receive a great deal of attention from the community due to its clear importance, especially in aerospace applications (see, e.g. [3] and references therein). In this paper, the focus is on the problem of Fault Detection and Diagnosis (FDD), which consists of reconstructing the faults affecting the system. This allows the designer to take fault levels/signals into account for the design of fault-tolerant controllers, thus achieving more resilience to faulty conditions. The design of FDD algorithms for multicopters using different strategies has been considered in [4–8]. In this paper, the technique considered for fault reconstruction is based on sliding mode observers [9].

Sliding mode observers are a particular type of Variable Structure Control Systems (VSCS), which are systems characterized by the presence of feedback control laws together with a decision rule [10]. Sliding mode observers attempt to estimate the system's states by ensuring that the observer's dynamics slide on a surface where the output estimation error is zero. This allows the observer to estimate the plant's internal states

while tracking its measured output, with the added advantage of achieving finite-time convergence of the state estimation error. The use of the equivalent output injection signal to reconstruct actuator and sensor faults was proposed in [11] and has been exploited in different applications [12–14]. This approach was later extended to account for uncertainties in [15] and to tackle linear parameter-varying (LPV) systems [16]. The case of simultaneous actuator and sensor faults was considered in [17–19]. The reconstruction of simultaneous actuator/sensor faults using other methods has also been reported in the literature (see, e.g. [20–22]). The application of this class of sliding mode observers for FDD in multicopters was considered in [23]. Flight tests of sliding mode FDD and FTC schemes have been reported in [24, 25].

Despite the increased interest for FTC work especially using SMC schemes, most of the existing work concentrated on simulation. There is a small number of works that consider implementation on actual hardware (see, e.g. [24, 25]). However, most of these works consider bespoke or expensive drones and lab setup which is not available to most researchers. Furthermore, in most countries, there are restrictions and regulations with regards to flight tests, especially for drones above certain weight.

The inspiration of this paper is to mitigate the two issues relating to cost and restrictions with regards to implementation work of state of the art control schemes (especially FTC and FDD). In this paper, a sliding mode observer is designed to provide fault detection and diagnosis for the Parrot® minidrone. The Parrot minidrone is a low-cost drone (under £100) and weighs only 68 g (thus avoiding flight restrictions due to its relative small size and weight). The main contribution of this paper is the implementation and flight testing of an LPV sliding mode observer on an actual minidrone, with an assessment of its in-flight FDD capabilities for simultaneous actuator and sensor faults. The motivations of this paper are (1) to show that the simultaneous actuator and sensor FDD scheme have low computational load and is therefore implementable on the on-board processor of the Parrot minidrone (which has limited processing power); (2) to show that implementation and flight tests of state-of-the-art control schemes do not require expensive setup (such as motion capture cameras and bespoke drones) and can be implemented almost anywhere. for example, in a small confined space. The implementation work using the Parrot minidrone also take advantage of the official Matlab/Simulink support which allows rapid prototyping of the control schemes to be implemented quickly without the inconvenience of writing the code from scratch. The results show that the observer is able to estimate the system outputs with high precision while also reconstructing actuator and sensor faults with good accuracy.

The paper is structured as follows. Section 2 gives a description for the Parrot® Rolling Spider minidrone, with the updated firmware and control structure used in the flight tests. The observer design is discussed in Section 3, beginning with the model description and ending with the numerical implementation. Finally, Section 4 presents some simulation and flight test results obtained with the minidrone.

FIGURE 1 Parrot® rolling spider minidrone [26]

2 | ROLLING SPIDER MINIDRONE

The Parrot® Rolling Spider is a small quadrotor developed for educational and recreational purposes (see Figure 1). Its small size allows it to be flown in confined spaces and ensures that it is below the minimum weight threshold where pilot registration is required in many countries. Its educational vocation was consolidated with the development of a hardware support add-on for Matlab/Simulink [27], based on a toolbox developed at the Massachusetts Institute of Technology [28]. The add-on allows the design and simulation of algorithms for flight control, estimation and observation using a nonlinear model of the minidrone in Simulink. The developed algorithms can then be easily deployed to the minidrone hardware via a Bluetooth connection and tested in actual flight.

The Rolling Spider gains its name from the pair of wheels that can be attached to it (see Figure 1). Their purpose is to protect the drone from physical shocks, to prevent the blades from causing harm, as well as allowing the minidrone to roll above or below surfaces. The minidrone alone weighs around 55 g, and 68 g with its wheels attached. Each of the motors provides 33 g of thrust, for a total of 132 g. It is controlled by an Arm Cortex-A9, clocked at 800 MHz, that runs an embedded Linux system [28] (as a comparison, a Raspberry Pi has a 64 bit quad-core ARM Cortex A-72 processor, clocked at 1500 MHz, which provides at least twice the speed of the processor in the Parrot® drone). A 6-axis accelerometer/gyroscope serves as the inertial measurement unit (IMU), providing translational acceleration and angular velocity measurements. A pressure sensor and a sonar are used for altitude measurements. Finally, a 60-FPS downward-facing camera is used to provide optical flow measurements to estimate horizontal positions and velocities.

The Matlab/Simulink add-on [27] provides new firmware for the flight control computer that adds an interface allowing users to define custom flight control codes as well as new image processing capabilities and optical flow handling. The control algorithms are executed by the flight control computer with a sampling rate of 200 Hz, which means a sample time of 5 ms. The toolbox includes a framework for controller design and simulation with a nonlinear model in Simulink. A pre-defined

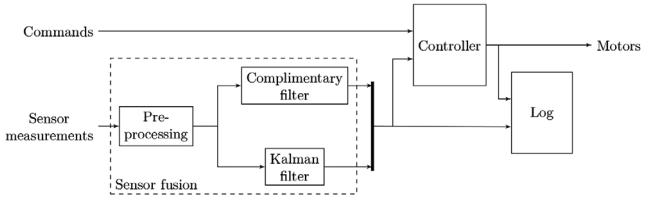


FIGURE 2 Simplified diagram of the controller structure [27]

controller structure is provided, based on the work reported in [29]. A simplified diagram of the structure is shown in Figure 2. Sensor fusion is present to provide estimates of the minidrone's states using the raw sensor readings. After a pre-processing, that handles the calibration offset and some preliminary filtering, the sensor measurements are fed to a complementary filter and to a Kalman filter [27, 29]. The former is used to provide estimates of attitude and the latter to provide estimates of position and velocities. The controller receives the resulting state from the sensor fusion block together with commands that are pre-programmed by the user (such as position/attitude references). The controller consists of a PID controller for pitch/roll control; a PD controller for yaw control and a PD controller for position control [27]. The controller provides a reference value for the desired rotation of each of the four rotors. The state estimates provided by the estimator and the controller output are logged in the internal memory of the minidrone, and can be accessed post-flight. A set of rules is present to handle errors during flight, such as a fault in the optical flow measurements, for example, as well as the take-off maneuver. The take-off maneuver consists of a 1 s period during which a fixed command 20% above hover thrust is given to each rotor and only the attitude control is active [28].

In this paper, the focus is on the development and flight evaluation of a sliding mode observer with the goal of reconstructing actuator and sensor faults. For this reason, the original sensor fusion block and the controller are not modified and are directly used in the flight tests, in order to simplify the test campaign. The output of the estimator will hence be used as the output of the minidrone for the purpose of testing the developed observer.

3 | MODELING AND OBSERVER DESIGN

3.1 | Model of the minidrone

In order to design the observer for the minidrone, a model of its dynamics is needed. A number of assumptions are made for this purpose [30]:

- the quadrotor is rigid and symmetric;
- the hub forces and rolling moments are neglected;
- the propellers are rigid;
- the ground effect is neglected;
- motors respond instantaneously.

Using the above (see, e.g. [30, 31]), a control-affine nonlinear model of the quadrotor can be obtained:

$$\dot{X}(t) = f_X(t,X) + g_X(t,X)\tau(t), \tag{1}$$

where the state X is given by

$$X = \begin{bmatrix} x & y & z & \phi & \theta & \psi & \dot{x} & \dot{y} & \dot{z} & p & q & r \end{bmatrix}^{\mathsf{T}}, \tag{2}$$

with x, y and z being the position along the body x, y, z-axes; ϕ, θ and ψ being the roll, pitch and yaw angles; $\dot{x}, \dot{y}$ and $\dot{z}$ being the velocity along the body x, y, z-axes; and p, q and r being the roll, pitch and yaw rates. The input signal $\tau(t)$ is given by

$$\tau(t) = \begin{bmatrix} \tau_z(t) & \tau_\phi(t) & \tau_\theta(t) & \tau_\psi(t) \end{bmatrix}^{\mathsf{T}}, \tag{3}$$

with τ_z representing the total thrust and $\tau_\phi, \tau_\theta, \tau_\psi$ the roll, pitch and yaw torques. The nonlinear functions $f_X(\cdot)$ and $g_X(\cdot)$ are given by

$$f_X(t,X) = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & a_\phi(X) & a_\theta(X) & a_\psi(X) & 0 & 0 & g & a_p(X) & a_q(X) & c_8 pq \end{bmatrix}^{\mathsf{T}}, \tag{4}$$

and

$$g_X(t,X) = \begin{bmatrix} 0_{6\times 6} & & & \\ c_1 b_x(X) & 0 & 0 & 0 \\ c_1 b_y(X) & 0 & 0 & 0 \\ -c_1 b_z(X) & 0 & 0 & 0 \\ 0 & c_4 & 0 & 0 \\ 0 & 0 & c_7 & 0 \\ 0 & 0 & 0 & c_9 \end{bmatrix}, \tag{5}$$

where

$$\begin{aligned} a_\phi(X) &:= p + q\sin(\phi)\tan(\theta) + r\cos(\phi)\tan(\theta) \\ a_\theta(X) &:= q\cos(\phi) - r\sin(\phi) \\ a_\psi(X) &:= q\sin(\phi)\sec(\theta) + r\cos(\phi)\sec(\theta) \\ a_p(X) &:= c_2 qr + c_3 \Omega_r(t) \\ a_q(X) &:= c_5 pr - c_6 \Omega_r(t) \\ b_x(X) &:= \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ b_y(X) &:= \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \\ b_z(X) &:= \cos(\phi)\cos(\theta). \end{aligned} \tag{6}$$

TABLE 1 Minidrone parameter definitions

Symbol	Parameter	Value	Unit
m	Mass	0.068	kg
I_{xx}	Inertia on the x-axis	6.86×10^{-5}	kg×m^2
I_{yy}	Inertia on the y-axis	9.20×10^{-5}	kg×m^2
I_{zz}	Inertia on the z-axis	1.37×10^{-4}	kg×m^2
b	Thrust factor	4.72×10^{-8}	N×s^2
d	Drag factor	1.139×10^{-10}	N×m×s^2
J_r	Rotor inertia	1.021×10^{-7}	kg×m^2
l	Moment arm length	4.412×10^{-2}	m

and the constant parameters c_1 to c_9 are given by

$$c_1 = \frac{1}{m} \quad c_2 = \frac{I_{yy} - I_{zz}}{I_{xx}} \quad c_3 = \frac{J_r}{I_{xx}} \quad c_4 = \frac{1}{I_{xx}}$$
$$c_5 = \frac{I_{zz} - I_{xx}}{I_{yy}} \quad c_6 = \frac{J_r}{I_{yy}} \quad c_7 = \frac{1}{I_{yy}} . \tag{7}$$
$$c_8 = \frac{I_{xx} - I_{yy}}{I_{zz}} \quad c_9 = \frac{1}{I_{zz}}$$

The variable $\Omega_r(t)$ is the residual propeller angular velocity, and is given by:

$$\Omega_r(t) := -\Omega_1(t) + \Omega_2(t) - \Omega_3(t) + \Omega_4(t), \tag{8}$$

where $\Omega_i(t)$ is the angular rate of the i^{th} motor. The definition of the remaining physical constants is given in Table 1 (in Section 3.1 below) along with the corresponding numerical values for the Parrot® minidrone [27].

For the purposes of the observer design, the nonlinear model of the quadrotor is rewritten as a linear parameter-varying (LPV) model. This greatly simplifies the design approach and allows the use of convex optimization methods with linear matrix inequality (LMI) constraints to provide a systematic observer synthesis procedure. In order to simplify the model and reduce the number of parameters, it is assumed that $(\dot{\phi}, \dot{\theta}, \dot{\psi}) \approx (p, q, r)$, that is, the deviations from hover are small [30, 31]. It is also assumed that the thrust and drag are proportional to the square of the rotor speed [30, 31]. The states x and y, along with their respective derivatives, are not included in the LPV model to reduce its complexity. This is also in accordance with standard practices for controller design, given that these states are normally controlled via a second (outer) loop (see, e.g. [31]). Based on the above, the quadrotor model can be written as

$$\begin{aligned} \dot{x}_p(t) &= A_p(\rho)x_p(t) + B_p(\rho)u(t) + M_p\xi_p(t, x_p) + N_g g, \\ y_p(t) &= C_p x_p(t) + d_p(t), \end{aligned} \tag{9}$$

where $A_p(\rho) \in \mathbb{R}^{8\times8}$ and $B_p(\rho) \in \mathbb{R}^{8\times4}$ are parameter-varying matrices, while $C_p \in \mathbb{R}^{7\times8}$, $M_p \in \mathbb{R}^{8\times2}$ and $N_g \in \mathbb{R}^{8\times1}$ are fixed matrices. The time-varying parameters

$$\rho(t) := [\rho_1 \quad \rho_2 \quad \rho_3 \quad \rho_4] = \begin{bmatrix} p(t) & q(t) & r(t) & b_{\dot{z}}(X) \end{bmatrix}, \tag{10}$$

are assumed to belong to a compact set $\Theta \subset \mathbb{R}^4$. The LPV reduced state vector is given by

$$x_p = \begin{bmatrix} z & \phi & \theta & \psi & \dot{z} & p & q & r \end{bmatrix}^\top \in \mathbb{R}^8, \tag{11}$$

and the control input u is given by

$$u = \begin{bmatrix} \Omega_1^2 & \Omega_2^2 & \Omega_3^2 & \Omega_4^2 \end{bmatrix}^\top \in \mathbb{R}^4, \tag{12}$$

with Ω_i representing the angular velocity of the i^{th} motor, and the measured output is

$$y_p = \begin{bmatrix} z & \phi & \theta & \psi & \dot{z} & q & r \end{bmatrix}^\top \in \mathbb{R}^7. \tag{13}$$

The parameter-varying state space matrices are given by

$$A_p(\rho) = \left[\begin{array}{c|c} 0_4 & I_4 \\ \hline 0_4 & A_{p,22}(\rho) \end{array}\right] \quad B_p(\rho) = \left[\begin{array}{c} 0_4 \\ \hline B_{p,2}(\rho) \end{array}\right],$$
$$C_p = \begin{bmatrix} I_5 & 0_{5\times2} \\ 0_{2\times5} & I_2 \end{bmatrix}, \tag{14}$$

with $A_{p,22}(\rho)$ and $B_{p,2}(\rho)$ defined as

$$A_{p,22}(\rho) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{c_2}{2}\rho_3 & \frac{c_2}{2}\rho_2 \\ 0 & \frac{c_5}{2}\rho_3 & 0 & \frac{c_5}{2}\rho_1 \\ 0 & \frac{c_8}{2}\rho_2 & \frac{c_8}{2}\rho_1 & 0 \end{bmatrix}, \tag{15}$$

$$B_{p,2}(\rho) = \begin{bmatrix} -bc_1\rho_4 & -bc_1\rho_4 & -bc_1\rho_4 & -bc_1\rho_4 \\ blc_4 & blc_4 & -blc_4 & -blc_4 \\ blc_7 & -blc_7 & -blc_7 & blc_7 \\ -dc_9 & dc_9 & -dc_9 & dc_9 \end{bmatrix}.$$

The signal $\xi_p(t, x_p) \in \mathbb{R}^2$ represents the uncertainty in the model. In the present case, it is assumed that the residual propeller angular speed $\Omega_r(t)$ is not perfectly known, and as such its effect is embedded in the uncertainty as

$$\xi_p(t) = \begin{bmatrix} c_3 q(t)\Omega_r(t) & -c_6 p(t)\Omega_r(t) \end{bmatrix}^\top, \tag{16}$$

and then M_p is defined as

$$M_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^\mathsf{T}. \tag{17}$$

The gravitational constant is denoted by g and the matrix N_g is simply used to distribute the gravitational action to the row corresponding to the state $\dot{z}$, and is thus defined as

$$N_g = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}. \tag{18}$$

Finally, $d_p(t) \in \mathbb{R}^7$ represents a corruption of the output signal due to imperfect measurement [16]. It is assumed to have a fixed bandwidth with a first-order low pass filter structure given by

$$\dot{d}_p(t) = -a_d d_p(t) + a_d \varphi(t), \tag{19}$$

where a_d is a positive scalar and φ is an unknown but bounded input signal. Both the uncertainty ξ_p and the disturbance d_p are assumed to be unknown but bounded. The physical parameters of the Rolling Spider minidrone can be found in [27] and are given in Table 1.

Note that the output matrix C_p is parameter independent, and has full row rank (meaning that no output is simply a linear combination of any other outputs). The input $u(t)$, the output $y_p(t)$ and the scheduling parameters $\rho(t)$ are assumed to be known and available to be used by the observer in real time.

3.2 | Observer design

This section describes the design of the LPV sliding mode observer for simultaneous actuator and sensor faults. The methodology is based on [17].

3.2.1 | Actuator and sensor faults

In this paper, it is assumed that the first and third rotors as well as the vertical velocity sensor ($\dot{z}$) are prone to faults. Based on this assumption, the LPV system (9) can be rewritten as

$$\begin{aligned} \dot{x}_p(t) &= A_p(\rho)x_p(t) + B_p(\rho)u(t) + H_p f_i(t) + M_p \xi_p(t, x_p) + N_g g, \\ y_p(t) &= C_p x_p(t) + N_p f_o(t) + d_p(t), \end{aligned} \tag{20}$$

with $H_p \in \mathbb{R}^{8\times 2}$ and $N_p \in \mathbb{R}^{7\times 1}$. The actuator and sensor faults are modeled as additive perturbations via the signals $f_i(t) \in \mathbb{R}^2$ and $f_o(t) \in \mathbb{R}$. Consequentially the fault-free conditions are given by $f_i \equiv 0$ and $f_o \equiv 0$. The actuator faults are assumed to be deviations from the nominal angular speed of the first and third motors of the quadrotor. This is modelled by choosing H_p as the first and third columns of $B_p(\rho)$. Since the term $bc_1\rho_4$ is small in comparison to the others, it is neglected for the purposes of the observer design, which yields

$$H_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & blc_4 & blc_7 & -dc_9 \\ 0 & 0 & 0 & 0 & 0 & -blc_4 & -blc_7 & -dc_9 \end{bmatrix}^\mathsf{T}. \tag{21}$$

Among the outputs of the model, the measurement of velocity along the body z-axis is assumed to be prone to faults. This output is chosen as it is estimated based on measurements of the position in the z-axis, and as such may be more sensitive to measurement errors. This choice is reflected in the model by setting

$$N_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}, \tag{22}$$

so that the output fault $f_o(t)$ is a scalar that only affects the output channel related to the state $\dot{z}$.

For the design of the observer, the number of fault-prone sensors ($\dot{z}$) is less than the number of outputs in (13). With a permutation of the output order, the following canonical form can be achieved:

$$\begin{aligned} \tilde{y}_p(t) &= \begin{bmatrix} y_{p,1}(t) \\ y_{p,2}(t) \end{bmatrix} \begin{matrix} \} \text{fault-free} \\ \} \text{prone to fault} \end{matrix} \\ &= \begin{bmatrix} C_{p,1} \\ C_{p,2} \end{bmatrix} x_p(t) + \begin{bmatrix} 0_{6\times 1} \\ 1 \end{bmatrix} f_o(t) + \begin{bmatrix} d_{p,1}(t) \\ d_{p,2}(t) \end{bmatrix}, \end{aligned} \tag{23}$$

with $y_{p,2} = \dot{z}$ and $y_{p,1}$ containing the remaining outputs.

3.2.2 | Augmented system

In order to be able to reconstruct sensor faults, the approach presented in [17] requires that the problem be recast as an input fault formulation. This is achieved by adding a first-order filter to the fault-prone output $y_{p,2}$, thus creating a new state $x_f(t) \in \mathbb{R}$ given (in this case) by

$$\dot{x}_f(t) = -A_f x_f(t) + A_f y_{p,2}(t), \tag{24}$$

with $A_f \in \mathbb{R}_+$. By augmenting the state x_p in (9) with x_f from (24) and using (23), the following augmented system is obtained:

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{x}_p(t) \\ \dot{x}_f(t) \end{bmatrix}}_{\dot{\bar{x}}(t)} &= \underbrace{\begin{bmatrix} A_p(\rho) & 0 \\ A_f C_{p,2} & -A_f \end{bmatrix}}_{\bar{A}(\rho)} \underbrace{\begin{bmatrix} x_p(t) \\ x_f(t) \end{bmatrix}}_{\bar{x}(t)} + \underbrace{\begin{bmatrix} B_p(\rho) \\ 0 \end{bmatrix}}_{\bar{B}(\rho)} u(t) \\ &+ \underbrace{\begin{bmatrix} H_p & 0 \\ 0 & A_f \end{bmatrix}}_{\bar{F}} \underbrace{\begin{bmatrix} f_i(t) \\ f_o(t) \end{bmatrix}}_{\bar{f}(t)} + \underbrace{\begin{bmatrix} M_p & 0 \\ 0 & A_f \end{bmatrix}}_{\bar{M}} \underbrace{\begin{bmatrix} \xi_p(t, x_p) \\ d_{p,2}(t) \end{bmatrix}}_{\bar{\xi}(t,\bar{x})} + \underbrace{\begin{bmatrix} N_g \\ 0 \end{bmatrix}}_{\bar{N}_g} g. \end{aligned} \tag{25}$$

The output of the augmented system is composed of the original fault-free output $y_{p,1}(t)$ and the filtered version of the fault-prone output $y_{p,2}$, given by $x_f(t)$:

$$\underbrace{\begin{bmatrix} y_{p,1}(t) \\ x_f(t) \end{bmatrix}}_{\bar{y}(t)} = \underbrace{\begin{bmatrix} C_{p,1} & 0 \\ 0 & I_1 \end{bmatrix}}_{\bar{C}} \bar{x}(t) + \underbrace{\begin{bmatrix} d_{p,1}(t) \\ 0 \end{bmatrix}}_{\bar{d}(t)}. \tag{26}$$

In this paper, the augmented state vector $\bar{x}$ has a dimension of 9 while the augmented output $\bar{y}$ is of the same dimension as y_p. The augmented fault $\bar{f}(t) \in \mathbb{R}^3$ now contains both actuator and sensors faults, while there are no longer faults acting in the output $\bar{y}$. It is assumed that the augmented fault signal $\bar{f}(t)$ is bounded, that is, there exists a known function $a : \mathbb{R}_+ \to \mathbb{R}_+$ such that

$$\left\| \bar{f}(t) \right\| \leq a(t), \quad \forall t \in \mathbb{R}_+. \tag{27}$$

Finally, the augmented uncertainty $\bar{\xi}(t, \bar{x})$ is of dimension 3.

3.2.3 | LPV sliding mode observer

For the design of the observer, it is required that the state-space representation of the augmented system be transformed into an output canonical form, which is possible here since $\bar{C}$ is a fixed matrix [16]. A condition for the existence of the required coordinate transformation is that $\text{rank}(C_{p,1}F) = 2$, which is satisfied in this case. In the new coordinates, given by $\bar{x}(t) \mapsto T_o\bar{x}(t) = \tilde{x}(t)$ as detailed in [16], the state-space equations become

$$\begin{aligned} \dot{\tilde{x}}(t) &= \tilde{A}(\rho)\tilde{x}(t) + \tilde{B}(\rho)u(t) + \tilde{F}\bar{f}(t) + \tilde{M}\bar{\xi}(t, \tilde{x}) + \tilde{N}_g g, \\ \tilde{y}(t) &= \tilde{C}\tilde{x}(t) + \bar{d}(t), \end{aligned} \tag{28}$$

and in this new representation, the output and the fault distribution matrices have the special structure:

$$\tilde{C} = \begin{bmatrix} 0_{7\times 2} & T \end{bmatrix} \quad \text{and} \quad \tilde{F} = \begin{bmatrix} 0_{2\times 3} \\ \hline F_2 \end{bmatrix} = \begin{bmatrix} 0_{2\times 3} \\ \hline 0_{4\times 3} \\ F_o \end{bmatrix}, \tag{29}$$

with $T \in \mathbb{R}^{7\times 7}$ an orthogonal matrix and $F_o \in \mathbb{R}^{3\times 3}$ a nonsingular matrix. The proposed LPV sliding mode observer has the following structure (in the new coordinate system $\tilde{x}$):

$$\begin{aligned} \dot{\hat{x}}(t) &= \tilde{A}(\rho)\hat{x}(t) + \tilde{B}(\rho)u(t) - \tilde{G}_l(\rho)e_y(t) + \tilde{G}_n\nu(t) + \tilde{N}_g g \\ \hat{y}(t) &= \tilde{C}\hat{x}(t), \end{aligned} \tag{30}$$

where $\hat{x}(t) \in \mathbb{R}^9$ is the estimated state and $e_y(t) \in \mathbb{R}^7 := \hat{y}(t) - \bar{y}(t)$ is the output estimation error. The matrices $\tilde{G}_l(\rho), \tilde{G}_n \in \mathbb{R}^{9\times 7}$ are the observer gains that will be designed. Finally, the term $\nu(t) \in \mathbb{R}^7$ is given by

$$\nu(t) = -\mathcal{K}\frac{P_o e_y}{\left\| P_o e_y \right\|}, \tag{31}$$

which represents the nonlinear injection signal that is used to induce a sliding motion on the surface given by

$$\mathcal{S} = \{\tilde{e} \in \mathbb{R}^9 \mid \tilde{C}\tilde{e} = 0\}, \tag{32}$$

in the state estimation error space. The matrix $P_o \in \mathbb{R}^{7\times 7}$ is a symmetric positive definite matrix satisfying the Lyapunov equation

$$(\tilde{A}_{22}^s)^\mathsf{T} P_o + P_o\tilde{A}_{22}^s = -I_7, \tag{33}$$

where $\tilde{A}_{22}^s \in \mathbb{R}^{7\times 7}$ is a stable matrix that is used as design freedom. In [16], it is shown that selecting the scalar gain $\mathcal{K}$ such that

$$\mathcal{K} > \left\| \overline{CF} \right\| a(t) + \eta_0, \tag{34}$$

with η_0 a positive scalar and where $a(t)$ is the bound on the augmented fault in (27), ensures that sliding is maintained on the surface $\mathcal{S}$, despite the presence of faults and uncertainties.

Let the state matrix in (30) be partitioned as

$$\tilde{A}(\rho) = \begin{bmatrix} \tilde{A}_{11}(\rho) & \tilde{A}_{12}(\rho) \\ \tilde{A}_{21}(\rho) & \tilde{A}_{22}(\rho) \end{bmatrix}, \tag{35}$$

with $\tilde{A}_{22}(\rho) \in \mathbb{R}^{7\times 7}$. The linear and nonlinear gains $\tilde{G}_l(\rho)$ and $\tilde{G}_n$ have the following structure [16]:

$$\tilde{G}_l(\rho) = T_L^{-1}\begin{bmatrix} \tilde{A}_{12}(\rho)T^\mathsf{T} + L\tilde{A}_{22}(\rho)T^\mathsf{T} - \tilde{A}_{11}(\rho)T^\mathsf{T}L \\ T\tilde{A}_{22}(\rho)T^\mathsf{T} - T\tilde{A}_{21}(\rho)T^\mathsf{T}L - A_{22}^s \end{bmatrix}, \tag{36}$$

$$\tilde{G}_n(\rho) = T_L^{-1}\begin{bmatrix} 0_{2\times 7} \\ I_7 \end{bmatrix}, \tag{37}$$

where $T_L \in \mathbb{R}^{9\times 9}$ is given by

$$T_L := \begin{bmatrix} I_2 & L \\ 0_{7\times 2} & T \end{bmatrix}, \tag{38}$$

with T given in (29) and

$$L = \begin{bmatrix} L^0 & 0_{2\times 3} \end{bmatrix} \in \mathbb{R}^{2\times 7}. \tag{39}$$

The matrix $L^0 \in \mathbb{R}^{2\times 4}$ is design freedom that will be leveraged to ensure a stable sliding motion and to reduce the effect of the uncertainties $\overline{\xi}(t)$ and the measurement disturbance $\overline{d}(t)$ on the fault reconstruction performance [15, 16]. The fault reconstruction is obtained through the notion of the equivalent output injection $\nu_{eq}(t)$, which can be understood as the average instantaneous level that the nonlinear injection $\nu(t)$ has to take in order to maintain sliding [32]. In [11], it is argued that, during perfect sliding, $\nu_{eq} \to F_2\overline{f}(t)$. This suggests that the equivalent output injection can be used to estimate the aggregated fault $\overline{f}(t)$. In [15, 16], the proposed reconstruction signal is given by

$$\hat{f}(t) = WT^{\mathsf{T}}\nu_{eq}(t), \tag{40}$$

where $W \in \mathbb{R}^{3\times 7}$ is structured as

$$W = \begin{bmatrix} W_1 & F_0^{-1} \end{bmatrix}, \tag{41}$$

with $W_1 \in \mathbb{R}^{3\times 4}$ being design freedom and F_o defined in (29).

The goal of the design is twofold: synthesize an observer via the choice of L^0 and W_1 that is able to estimate the minidrone's states and reconstruct actuator and sensor faults while minimizing the effect of uncertainties in the fault reconstruction. In order to understand how that can be achieved, arguing as in [16], the operator mapping the uncertainties to the fault reconstruction error can be written as

$$\begin{aligned} \dot{e}_a(t) &= \mathcal{A}_a(\rho)e_a(t) + B_a\xi_a(t,\tilde{x}), \\ e_f(t) &= C_a e_a(t) + F_a\xi_a(t,\tilde{x}), \end{aligned} \tag{42}$$

where $e_a(t) := \mathrm{col}(\hat{d}_p(t), e_1(t)) \in \mathbb{R}^9$, with $\hat{d}_p(t) := T^{\mathsf{T}}d_p(t)$ and $e_1(t)$ the first two elements of the state estimation error $\hat{x}(t) - \tilde{x}(t)$; $e_f(t) := \hat{f}(t) - \overline{f}(t) \in \mathbb{R}^3$ is the fault reconstruction error and $\xi_a(t,\tilde{x}) := \mathrm{col}(\overline{\xi}(t,\tilde{x}), \hat{\varphi}(t)) \in \mathbb{R}^{10}$, with $\hat{\varphi} := T^{\mathsf{T}}\varphi(t)$ (where φ is defined in (19)). The matrices $\mathcal{A}_a(\rho), B_a, C_a(\rho)$ and F_a are given by

$$\begin{aligned} \mathcal{A}_a(\rho) &= \begin{bmatrix} -a_d I_7 & 0 \\ \tilde{\mathcal{A}}_{12}(\rho) + L\tilde{\mathcal{A}}_{22}(\rho) + a_d L & \tilde{\mathcal{A}}_{11}(\rho) + L\tilde{\mathcal{A}}_{21}(\rho) \end{bmatrix}, \\ B_a &= \begin{bmatrix} 0 & a_d I_7 \\ -(M_1 + LM_2) & -a_d L \end{bmatrix}, \\ C_a(\rho) &= \begin{bmatrix} -W\tilde{\mathcal{A}}_{22}(\rho) - a_d W & -W\tilde{\mathcal{A}}_{21}(\rho) \end{bmatrix}, \\ F_a &= \begin{bmatrix} WM_2 & a_d W \end{bmatrix}, \end{aligned} \tag{43}$$

where $M_1 \in \mathbb{R}^{2\times 3}$ and $M_2 \in \mathbb{R}^{7\times 3}$ are defined by the partition

$$\overline{M} = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}. \tag{44}$$

Since both the uncertainty $\overline{\xi}(t,\tilde{x})$ and the signal $\varphi(t)$ are assumed to be bounded, the system (42) describes an LPV operator from $\mathcal{L}_{2e}^{10}(\mathbb{R}_+)$ into $\mathcal{L}_{2e}^{3}(\mathbb{R}_+)$. The operator is stable provided that $\tilde{\mathcal{A}}_{11}(\rho) + L\tilde{\mathcal{A}}_{21}(\rho)$ is stable via an appropriate choice of L. Given that $\tilde{\mathcal{A}}(\rho)$ depends affinely on the scheduling parameters ρ and that these are bounded, an equivalent polytopic representation of the LPV operator can be found by using the 2^4 vertices of Θ. Let ρ_i denote the i^{th} vertex of the polytope defined by Θ, with $i \in \mathcal{I} := \{1, \dots, 2^4\}$. Define $\mathcal{A}_{a,i}$ and $C_{a,i}$ as fixed matrices obtained by replacing ρ by ρ_i, that is, $\mathcal{A}_{a,i} = \mathcal{A}_a(\rho_i)$ and $C_{a,i} = C_a(\rho_i)$. Using the bounded real lemma [16], an upper bound $\gamma > 0$ on the $\mathcal{L}_2$-gain from ξ_a to e_f can be computed provided there exist matrices $L^0 \in \mathbb{R}^{2\times 4}$, $W_1 \in \mathbb{R}^{3\times 4}$, $P_{a_d} \in \mathbb{R}^{7\times 7}$ and $P_{11} \in \mathbb{R}^{2\times 2}$ such that the following problem can be solved:

Minimize γ

subject to:

$$\begin{cases} \begin{bmatrix} \mathcal{A}_{a,i}^{\mathsf{T}}P_1 + P_1\mathcal{A}_{a,i} & P_1(B_a\Delta) & C_{a,i}^{\mathsf{T}} \\ (B_a\Delta)^{\mathsf{T}}P_1 & -\gamma I_{10} & (F_a\Delta)^{\mathsf{T}} \\ C_{a,i} & (F_a\Delta) & -\gamma I_3 \end{bmatrix} \prec 0, \quad \forall i \in \mathcal{I}. \\ P_1 = \begin{bmatrix} P_{a_d} & 0 \\ 0 & P_{11} \end{bmatrix} \succ 0 \end{cases} \tag{45}$$

In (45), the matrix $\Delta := \mathrm{diag}(\delta_1, \delta_2)$, where $\delta_1 \in \mathbb{R}^{3\times 3}$ and $\delta_2 \in \mathbb{R}^{7\times 7}$ are diagonal matrices with positive elements. These constitute a multiplicative weight to B_a and F_a to allow the designer to tune the relative importance of the uncertainties $\overline{\xi}(t,\tilde{x})$ and the measurement corruption $\hat{d}_p(t)$. Crucially, problem (45) is a convex optimization problem with linear matrix inequality constraints [33], for which there are several commercially available efficient solvers.

3.2.4 | Design summary

In this section, a brief summary of the design procedure is provided.

1. Write the system dynamics in the LPV form of (9);
2. Select which actuators and sensors are prone to faults, and permute the outputs so that the potentially fault-prone sensors are the last elements of the output vector as in (23);
3. Select a matrix $\mathcal{A}_f$ to filter the fault-prone sensors and obtain $\varkappa_f(t)$ in (24)—a filtered version of the output. Augment the LPV system with these filtered states to obtain the augmented representation (25) and (26);
4. If necessary (and provided that $\mathrm{rank}(C_{p,1}F) = 2$), permute the states and outputs to obtain the output canonical representation where (29) is satisfied;

5. Select a matrix $\mathcal{A}_{22}^s$ in view of the desired dynamics before sliding is attained, and solve (33) to obtain P_o;
6. Choose an appropriate value for a_d in view of the characterization of the measurement corruption $d_p(t)$;
7. Choose appropriate diagonal entries for matrices δ_1 and δ_2 to form the weighting matrix Δ;
8. Select the compact subset Θ defining the validity of the LPV model. Obtain from Θ a polytopic representation of the LPV system with vertices in $\mathcal{I}$.
9. Solve the optimization problem (45) with an appropriate numerical solver.
10. Obtain the linear and nonlinear gains $\tilde{G}_l(\rho)$ and $\tilde{G}_n$ as well as the reconstruction matrix W based on the optimization solution.
11. Select an appropriate value for the injection gain $\mathcal{K}$ in view of (27);
12. Simulate the observer, and fine tune the design by changing $\mathcal{A}_{22}^s, \mathcal{A}_f, a_d, \Delta$ and $\mathcal{K}$ if necessary.
(**Remark:** the selection of tuning parameters in step 12 will be described in detail in the subsequent subsection.)

3.3 | Numerical synthesis and implementation

In this section, the synthesis of the observer described in the previous section is described for the Parrot® minidrone. First, a limit of validity for the LPV model must be established via the bounded set Θ for the resolution of Problem (45). Since the construction of the LPV model is based on the assumption that the perturbations from hover are small, the minimum and maximum bounds for $p(t), q(t)$ and $r(t)$ are selected as ± 1 rad/s. Together with the natural bound on the term $b_{\tilde{z}}(X)$, this yields $\Theta = \{\rho_i \in [-1, 1], \text{ for } i = 1, \dots, 4\}$.

The LPV system is put in the canonical form (29) by applying the coordinate transformation $x(t) \mapsto T_o x(t)$ and the output permutation $\tilde{y}(t) = T_1 y(t)$, with

$$T_o = \begin{bmatrix} 0 & I_2 & 0 \\ I_4 & 0 & 0 \\ 0 & 0 & I_2 \end{bmatrix} \quad \text{and} \quad T_1 = \begin{bmatrix} I_4 & 0 & 0 \\ 0 & 0 & I_2 \\ 0 & 1 & 0 \end{bmatrix}. \tag{46}$$

Here, the filter matrix $\mathcal{A}_f$ in (24) is selected as $\mathcal{A}_f = 1$. As discussed in [34], selecting $\mathcal{A}_f$ with large diagonal values increases the bandwidth of the output filter, and thus improves the state estimation performance, while smaller values may improve (reduce) the upper bound γ on the $\mathcal{L}_2$-gain from the uncertainties to the fault reconstruction error. The latter is the focus in this paper, and hence this is reflected in the tuning of the $\mathcal{A}_f$ parameter. The design matrix $\mathcal{A}_{22}^s$ is chosen as $-10I_7$ to help ensure that sliding is achieved in a short time. The parameter a_d in (19) can also be used by the designer as a "tuning knob". After tuning, here its value is chosen as $a_d = 0.01$. This characterizes a signal $d_p(t)$ with a short bandwidth, but does not affect the performance of the system. Finally, the scaling weights are chosen as $\delta_1 = \text{diag}(1, 1, 50) \times 10^{-4}$ and $\delta_2 = \text{diag}(1, 1, 1, 1, 1, 1, 1) \times 10^{-4}$. This choice puts more weight into the output fault reconstruction by giving more importance to the channels related to the sensor fault.

TABLE 2 Flight test description matrix

Case			
Simul.	**Flight**	**Maneuver**	**Fault**
S0	F0	Step in x	No faults
	F1	Ramp in x and step in z	Ramp fault in the sensor
	F2	Triangle in x	Loss of efficiency fault in both actuators
	F3	Ramp in ψ	Step faults in both actuators and in the sensor
	F4	Triangle in y	Sine-wave fault in motor 3 and in the sensor

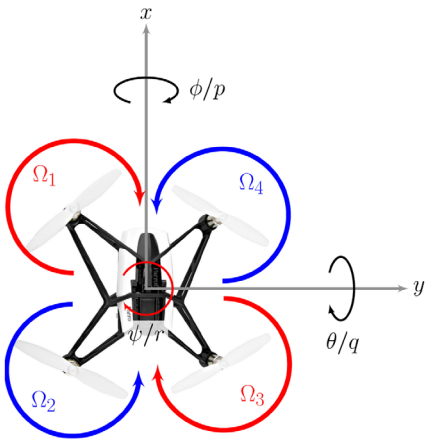


FIGURE 3 Body coordinate frame and motor rotation

The observer is synthesized in Matlab, where Problem (45) is coded by means of the YALMIP parser [35] and numerically solved with the MOSEK solver [36]. The numerical solution yields $\gamma = 1.9598$, and the design matrices L^0 and W^1 are given by

$$L^0 = \begin{bmatrix} 0.5454 & 1.8202 \times 10^{-5} \\ 1.4595 \times 10^{-6} & 6.6656 \\ 1.9579 \times 10^{-6} & 8.8527 \\ 0 & 5.4673 \times 10^{-6} \end{bmatrix}^{\mathsf{T}},$$

$$W^1 = \begin{bmatrix} -9.3471 \times 10^{-6} & 4.2673 \times 10^{-6} & -0.9990 \\ 1.4210 & -1.4346 & 1.7172 \times 10^{-3} \\ -1.8756 & 1.9048 & -3.8043 \times 10^{-3} \\ -7.6901 \times 10^{-2} & -7.6394 \times 10^{-2} & -3.0764 \times 10^{-4} \end{bmatrix}^{\mathsf{T}}. \tag{47}$$

The linear and nonlinear gains $\tilde{G}_l(\rho)$ and $\tilde{G}_n$, computed as in (36) and (37) and represented in the output canonical form, are given by (48)

$$\tilde{G}_l(\rho) = \begin{bmatrix} 5.4540 & 1.4595\times10^{-5} & 1.9579\times10^{-5} & 0 & 0 & 0 & 0 \\ 0.00018202 & 66.656 & 88.527 & 5.4673\times10^{-5} & 0.32507\rho_3+13.411 & 0.32507\rho_2 & 0 \\ 10.545 & 1.4595\times10^{-6} & 1.9579\times10^{-6} & 0 & 0 & 0 & 0 \\ 1.8202\times10^{-5} & 16.666 & 8.8527 & 5.4673\times10^{-6} & 1.3411 & 0 & 0 \\ 0 & 0 & 10 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 10.000 & 0 & 1 & 0 \\ 6.7269\times10^{-6}\rho_3 & 2.4634\rho_3 & 3.2717\rho_3 & 2.0205\times10^{-6}\rho_3 & 0.49563\rho_3+10 & 0.36957\rho_1 & 0 \\ 1.5591\times10^{-6}\rho_2 & 0.57092\rho_2 & 0.75825\rho_2 & 4.6829\times10^{-7}\rho_2 & 0.085652\rho_1-0.11487\rho_2 & 10 & 0 \\ 0.54540 & 1.4595\times10^{-6} & 1.9579\times10^{-6} & 0 & 0 & 0 & 9 \end{bmatrix} \tag{48}$$

and

$$\tilde{G}_n = \begin{bmatrix} -L \\ I_7 \end{bmatrix}, \tag{49}$$

with L given in (39). Finally, the matrix P_o is given by $P_o = 0.5I_7$.

For practical implementation, the nonlinear injection term (31) is replaced by a smooth approximation with the goal of reducing chattering. In this paper, a sigmoidal approximation is used:

$$\nu_\delta(t) = -\mathcal{K}\frac{P_o e_y(t)}{\left\|P_o e_y(t)\right\| + \delta_0}, \tag{50}$$

with δ_0 a small positive scalar so that ν depends smoothly on e_y. It is clear that $\nu_\delta \to \nu$ as $\delta_0 \to 0$, and thus δ_0 is chosen to achieve a compromise between chattering and the absence of sliding. In this paper, a value of $\delta_0 = 0.1$ is considered (after some tuning). The modulation gain $\mathcal{K}$ is chosen as $\mathcal{K} = 750$ to ensure that the observer is able to stay close to sliding even in the presence of faults.

The observer is implemented in the flight control computer of the Parrot® minidrone. As discussed in Section 2, the custom firmware of the Parrot® minidrone runs the control algorithm at a sampling rate of 200 Hz. For this reason, the LPV observer has been discretized in order to be implemented in the hardware. A simple discretization using the Forward Euler method [37] is used to limit the computation complexity at every step. The applicability of this discretization method was verified in simulations, which led to re-tuning of the observer parameters δ_0 and $\mathcal{K}$ (the values given above are the final values used in the hardware). The simulations in discrete time showed that the performance was very close to what was assessed in continuous time. A simulation scenario is presented in the next section.

4 | SIMULATION AND FLIGHT TEST RESULTS

In this section, two sets of verification results will be discussed. The first set is based on simulations while the second set is obtained in flight tests. Table 2 shows the different sets of results (simulations and flight tests) considered in this paper. One simulation test (S0) and four flight tests (F0–F4) are performed. The simulation test is performed in similar conditions (with respect to the maneuver and faults considered) to one of the flight tests, and the numeration of the simulation test indicates to which flight test it corresponds. A description for both sets of results is given in the following sections.

4.1 | Simulation

In this section the simulation results obtained using the designed observer are presented. The simulations are done using the flight simulation model provided in [27], which contains a nonlinear model of the minidrone including rotor and sensor dynamics. The test case is selected to correspond to one of the flight test scenarios, see Table 2. As the paper focuses on the quality of the simultaneous sensor and actuator fault estimation when implemented on the flight control computer, the existing controller was used to provide basic control without any fault tolerant capabilities. For this reason, the faults were only implemented at a software level before they were supplied to the observer, and thus do not affect the controller feedback loop (see Figure 4). This is done to maintain stability and control performance of the minidrone, thus ensuring safety during the flight tests and reducing the risk of damage to the minidrone. In order to be consistent with the discussion in the previous section, the actuator faults are added with an inverse sign. From the

point of view of the observer, both the online and offline cases are strictly equivalent.

The reconstructed sensor fault can be directly related to the additive fault in the output signal. Since the actuator fault is modeled as an additive signal added to the rotation of the rotors, the actual fault seen by the observer must account for the fact that the model input is the squared angular velocity of the rotors. Letting u_c be the command signal at the output of the controller and f_i the additive input fault, the faulty command sent to the observer is given by

$$\begin{aligned} u(t) &= (u_c(t) - f_i(t))^2 \\ &= u_c(t)^2 + (f_i(t)^2 - 2u_c(t)f_i(t)). \end{aligned} \tag{51}$$

This means that the observer is actually estimating the signal $\breve{f}_i(t) := f_i(t)^2 - 2u_c(t)f_i(t)$. An estimation of the actual additive fault $f_i(t)$, denoted $\hat{f}_i(t)$, can be recovered by means of the relation $\hat{f}_i(t) = u_c(t) - \sqrt{u_c(t)^2 - \breve{f}_i(t)}$, provided that $\breve{f}_i(t) \leq u_c(t)^2$.

The scenario considered in simulation is the nominal fault-free case. The minidrone takes off from the ground to hover at an altitude of 0.5 m, and a step of 2 m is sent to the x reference signal at 2 s. The 3-dimensional trajectory of the minidrone is shown in Figure 5a. The references given to the controller are shown in black dotted lines in the rightmost plots. The minidrone starts at the ground (red dot) and takes off to reach hover at an altitude of 0.5 m. The line color is used to represent the flow of time during the test, as indicated in the colorbar, going from light (begin) to dark (end) color. As discussed in Section 2, the controller implemented in [27] is programmed such that a dedicated take-off maneuver takes place during the first second of flight. It can be seen that some overshoot is present in the x and z channels due to the controller action. As described earlier, this is the performance of the existing controller and no modification was made. As the focus of this paper is to evaluate the performance of the simultaneous fault reconstruction observer when implemented on the flight controller, the existing controller was used to provide some level of control.

The outputs of the sensor fusion ($\bar{y}(t)$, in red solid lines) and of the observer ($\tilde{y}(t)$, in blue dashed lines) are shown in Figure 5c. The z and x_f plots are shown with opposed sign, so that positive values represent the altitude of the drone with respect to the ground. Figure 5c shows that the observer accurately tracks the system's output, with the exception of a small

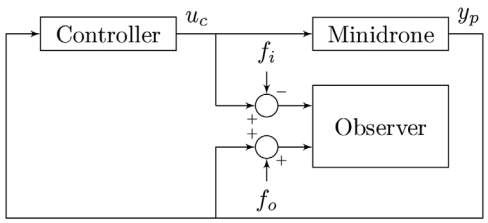


FIGURE 4 Actuator and sensor fault implementation on the minidrone

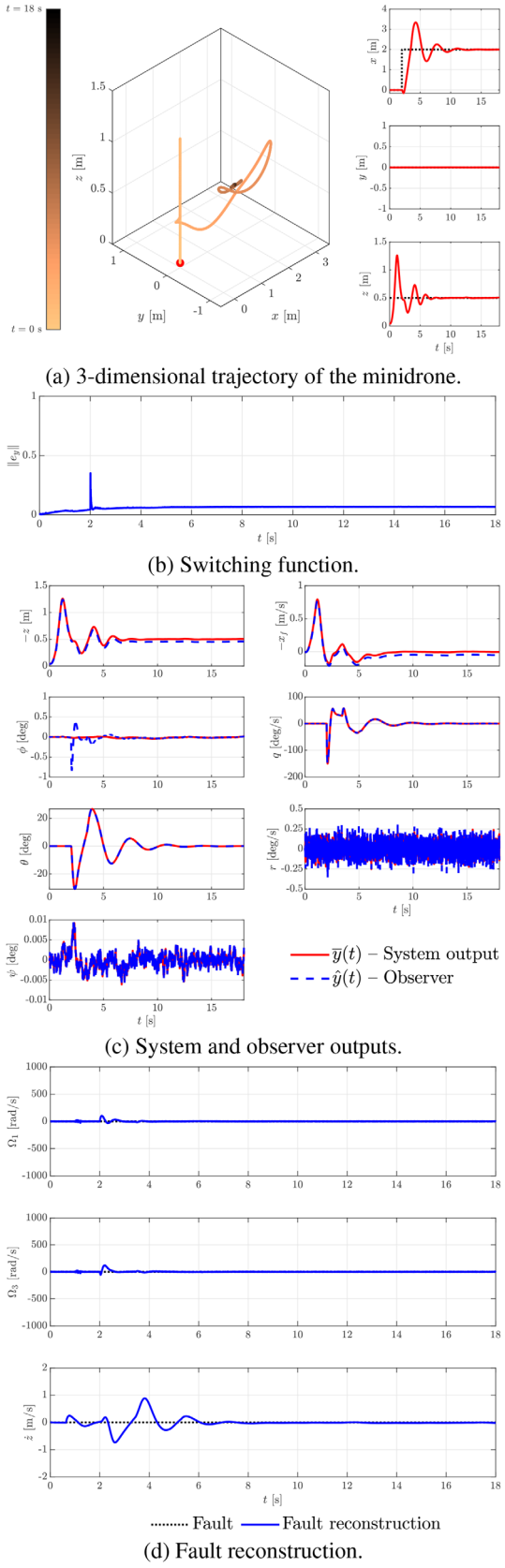


FIGURE 5 Case S0

spike seen in the ϕ channel. This could be due to the aggressiveness of the maneuvers at the start of the simulation (an increase in altitude followed by forward motion in the x axis) within a short period of time. Nonetheless, the ripple remains below 0.5 degrees and is quickly subdued.

Figure 5b shows the norm of the switching signal $e_y(t)$. With the exception of a spike at 2 s, the switching signal has a low amplitude throughout the test, which indicates that sliding is taking place along the surface $e_y(t) = 0$. The spike corresponds to the difference between ϕ as given by the estimator and by the observer (see Figure 5c).

The actuator and sensor faults reconstructed by the observer are shown in Figure 5d. The actuator and sensor faults are represented by black dotted lines, and the fault reconstruction is shown in blue solid lines. The observer is able to correctly estimate the zero fault in the fault-free case, which is important when the minidrone is flying in nominal conditions. A small deviation can be seen in the sensor reconstruction during the maneuver, but it quickly subsides after a few seconds.

4.2 | Flight test results

The observer designed in the previous section is implemented in the minidrone to run in real time during flight. A total of five flight tests are performed to assess the performance of the fault reconstruction capabilities of the sliding mode LPV observer. All flight tests are performed indoors at a confined space of less than 2×2×3 m (therefore gust effect is minimal) at room temperature and with the minidrone taking off from the ground. The list of test cases considered in this paper is given in Table 2. The faults are added via software in the flight control code, to simulate the effect of actual faults acting on the minidrone. In this way, different fault profiles can be tested to assess the performance of the observer in different cases.

4.2.1 | Computational load

The ARM-core Cortex-A9 available in the Parrot minidrone has a capability to execute a total of 0.372 giga-floating point operations per second (GFLOPs) [38, 39]. As a comparison, a modern PC processor, for example, Core i7-10710U, can execute a total of 105.6 GFLOPs [40]. Using the tool proposed in [41], the number of floating point operations related to the observer reconstruction scheme that are executed on the processor at each iteration is estimated to be 4126. Divided by the sampling time of 5 ms, this yields a requirement of 8.25×10^{-4} GFLOPS, which translates to less than 1% of the total processing power of the processor. This indicates a low computational load of the proposed scheme, on a relatively low-cost processor.

4.2.2 | Case F0

The first flight test consists of a hovering maneuver with a 2 m step in the x reference command at 2 s, for a total flight time of 18 s (see Figure 6a). The same convention as in Section 4.1 is used for the plots, with the red dot showing the initial position and the gradient line indicating the 3-dimensional trajectory as time flows from beginning (light color) to end (dark color). The references given to the controller are shown in black dotted lines in the rightmost plots. No faults are introduced in this test, with the goal of assessing the performance of the observer in nominal conditions.

Figure 6c shows the evolution of the output of the system ($\bar{y}(t)$) and of the observer ($\hat{y}(t)$) during the test. The z and x_f plots are once again shown with opposed sign, so that positive values represent the height attained by the drone. The red solid lines represent the output of the minidrone, while the blue dashed lines are the outputs of the sliding mode observer. The plots show that the observer is able to track the output of the drone very accurately.

Figure 6b shows the norm of the output error signal e_y, which acts as the switching function of the sliding mode observer. This means that perfect sliding happens when $e_y(t) = 0$. The plot shows that the signal stays close to zero for the whole duration of the flight, and hence that sliding occurs. A small peak can be seen around the 2 s mark, which is due to the aggressive maneuver undertaken by the minidrone in response to the step in the x reference signal (see Figure 6a).

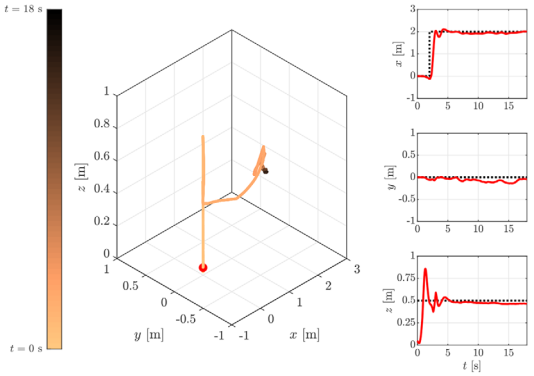
The actuator and sensor faults reconstructed by the observer are shown in Figure 6d. The plots show that the observer is able to correctly estimate the zero fault in the no-fault case, which is important when the minidrone is flying in nominal conditions.

4.2.3 | Case F1

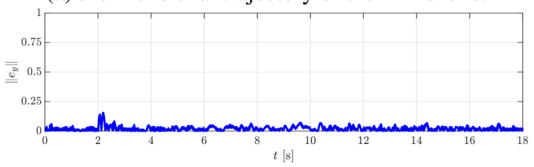
The second faulty scenario is considered in Case F1. This flight test evaluates the fault reconstruction performance when only a sensor fault is present. The 3-dimensional trajectory is shown in Figure 7a. The minidrone takes off from the ground to reach hover at an altitude of 0.5 m. After 2 s, a ramp reference signal is sent in the x direction, with a slope of 0.5 m/s. Finally, at 6 s, a step signal is sent in the z reference for the minidrone to reach an altitude of 1 m. A ramp fault is added to the $\dot{z}$ measurement signal starting at 3 s, with a slope of -0.25 m/s.

As in the previous cases, the outputs of the system and observer are compared in Figure 7c. The observer is once again capable of tracking the system outputs during the entire flight test. The effect of the ramp fault can be seen in the plot of $-x_f(t)$, in the upper right corner. Even in the presence of this fault, the output tracking remains satisfactory in all measured outputs.

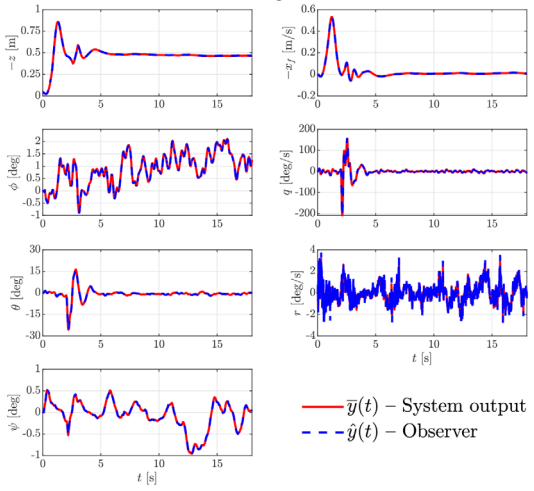
The norm of the switching function $e_y(t)$ is shown in Figure 7b. Despite some oscillation before 2 s, which can be associated with the take-off maneuver, the observer is able to maintain sliding throughout most of the flight test. The fault reconstruction by the observer is shown in Figure 7d. The plots show that the observer is able to estimate ramp faults quite accurately, without presenting false positives in the actuator fault reconstruction.



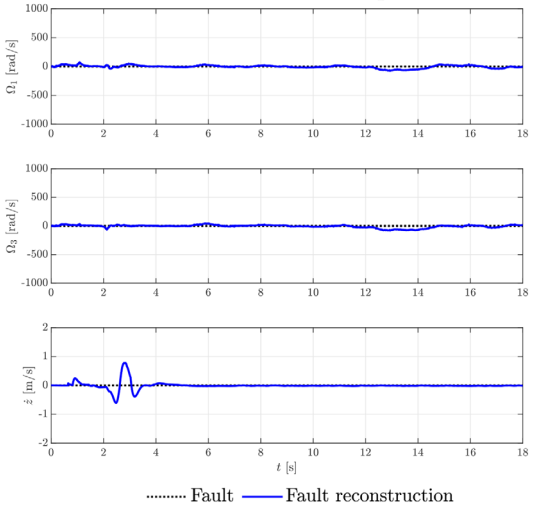
(a) 3-dimensional trajectory of the minidrone.



(b) Switching function.

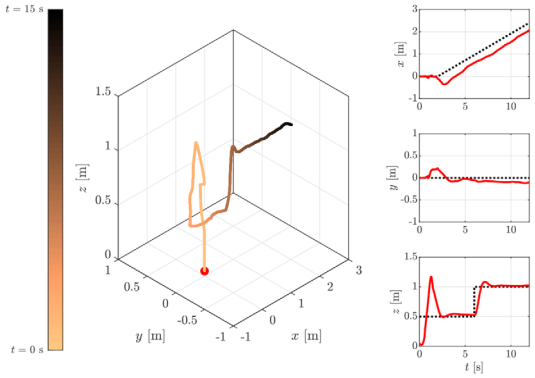


(c) System and observer outputs.

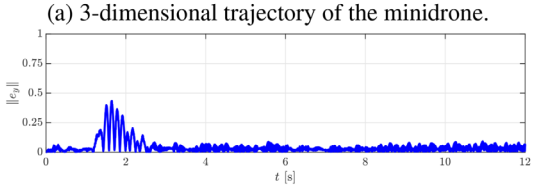


(d) Fault reconstruction.

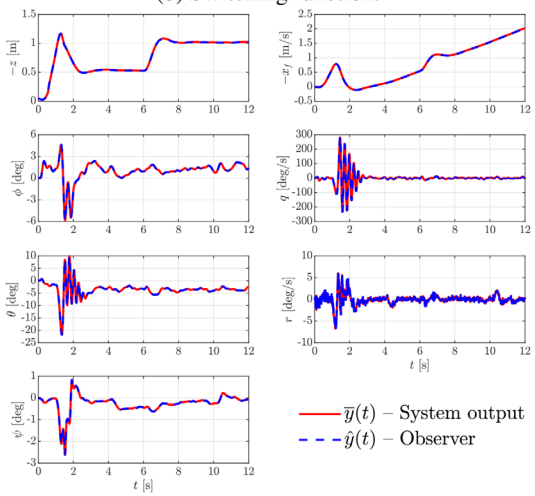
FIGURE 6 Case F0



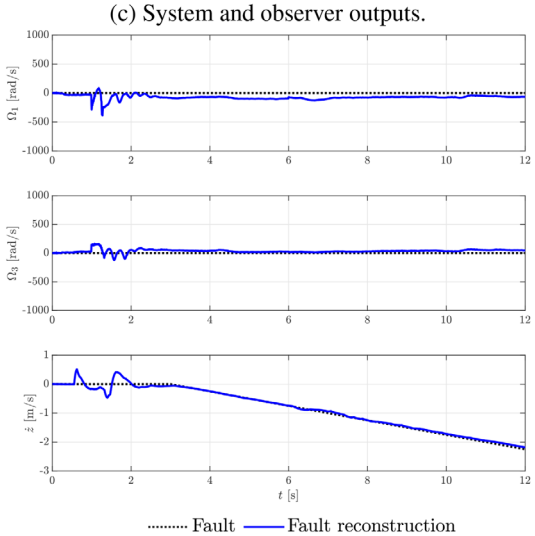
(a) 3-dimensional trajectory of the minidrone.



(b) Switching function.



(c) System and observer outputs.



(d) Fault reconstruction.

FIGURE 7 Case F1

4.2.4 | Case F2

The second faulty scenario consists of loss of efficiency faults in both actuators. The faults are activated at 2 s, and consist of a 40% loss of efficiency in both motors 1 and 3 (see Figure 8d). The maneuver consists of taking off from the ground to attain hover at 0.5 m altitude, with a triangle reference in the x direction that causes the minidrone to advance 2 m and come back. The 3-dimensional trajectory is shown in Figure 8a.

Figure 8c presents the comparison between the system outputs and the observer. The observer tracks the outputs with good accuracy, and both signals are mostly superposed for the duration of the test.

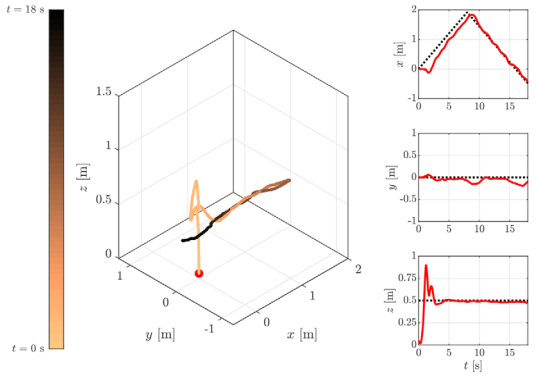
The norm of the switching signal $e_y(t)$ is presented in Figure 8b. In this flight test, $\|e_y(t)\|$ remains quite close to zero, meaning that sliding is maintained for the entire test duration.

The reconstructed faults are shown in Figure 8d. The added actuator faults, shown in black dotted lines, consist of 40% of the corresponding commanded values. During the maneuvers in this test, the commanded angular velocity of motors 1 and 3 remain relatively constant, as seen by the corresponding faults. The fault reconstruction signal is seen to once again track the faults with very good accuracy.

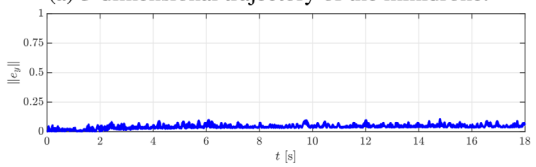
4.2.5 | Case F3

In the third flight test, simultaneous actuator and sensor faults are evaluated. The minidrone is set to take-off and hover at 1 m altitude with a ramp on the ψ reference signal. The 3-dimensional trajectory undertaken by the minidrone is shown in Figure 9a. An 800 rad/s step fault is added to each fault-prone actuator, the first in motor 1 at 2 s and the second in motor 3 at 4 s. A negative unit step is also added to the estimated value of $\dot{\hat{\chi}}$ at 2 s (see Figure 9d). Figure 9c shows the evolution of the output of the system (red solid lines) and of the observer (blue dashed lines). The performance of the observer is similar to what was seen in the fault-free case, with very accurate tracking of the estimator output signals.

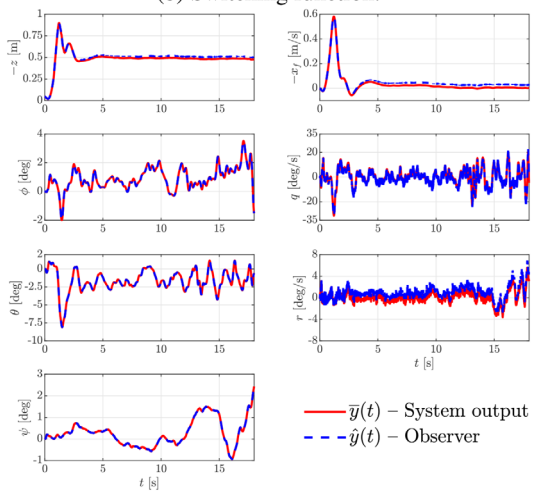
The switching function evolution is presented in Figure 9b. In contrast to Figure 6b (relative to the fault-free case), in this flight test the switching function is seen to deviate from the sliding surface between 2 and 4 s, that is, between the instants where the faults in motor 1 and 3 start. This is mostly due to a bias in the q output signal due to the fault in motor 1. Nonetheless, the deviation from zero remains reasonable and, as seen in Figure 9d below, the fault reconstruction is not affected. The added faults (black dotted lines) and the corresponding reconstruction given by the observer (blue solid lines) are shown in Figure 9d. The plots show that the observer is able to accurately identify concurrent faults in actuators and sensors, even when they take place at the same instant. The maximum theoretical angular velocity of the rotors is around 2631 rad/s, which means that the added actuator faults represent about 30% of the maximum rotation. The fault identification presents a rapid response, effectively tracking the step fault with a 250 ms response time.



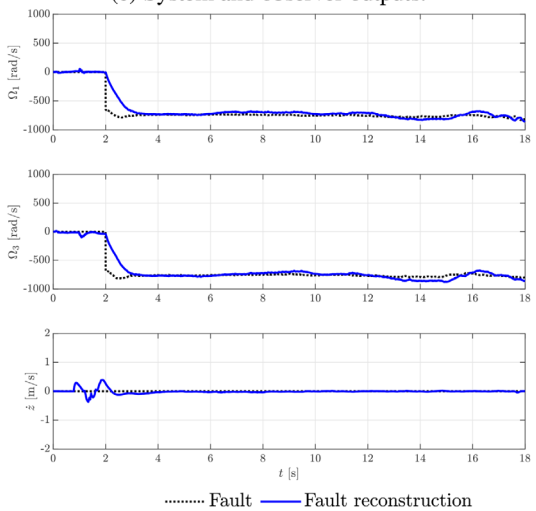
(a) 3-dimensional trajectory of the minidrone.



(b) Switching function.

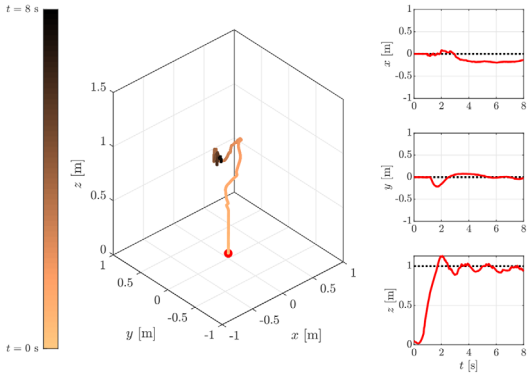


(c) System and observer outputs.

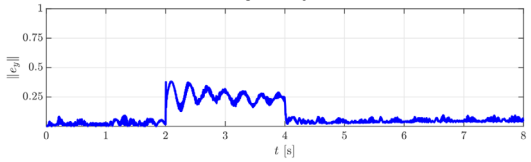


(d) Fault reconstruction.

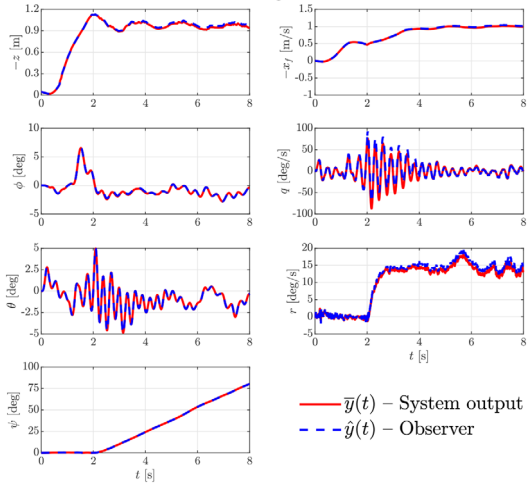
FIGURE 8 Case F2



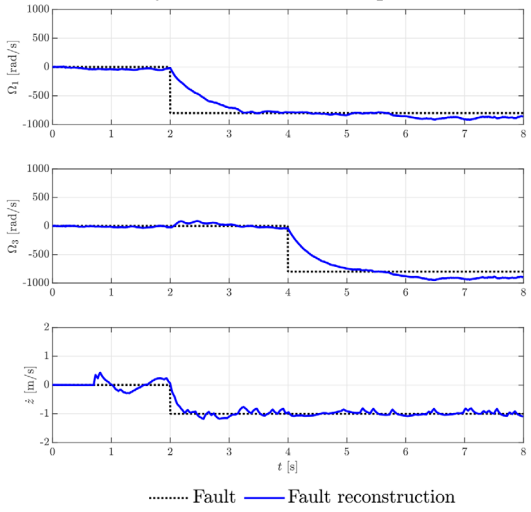
(a) 3-dimensional trajectory of the minidrone.



(b) Switching function.



(c) System and observer outputs.



(d) Fault reconstruction.

FIGURE 9 Case F3

As seen in the fault-free case, the fault reconstruction remains close to zero before the faults are active, which once again suggests that the observer does not misidentify false positives.

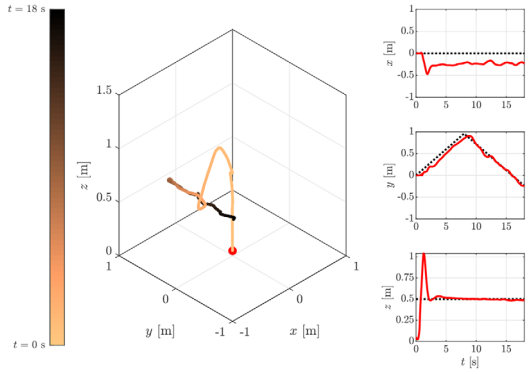
4.2.6 | Case F4

The final test considers simultaneous actuator and sensor fault reconstruction with sine wave faults. In this scenario, the minidrone reaches hover at an altitude of 0.5 m after take-off, and a triangle reference is given in y direction, causing the drone to move 2 m sideways and then come back to the original hovering point (see Figure 10a). The two actuators and the $\dot{\hat{x}}$ measurement are assumed to be faulty in this case, all with sinusoidal faults (see Figure 10d). The actuator fault consists of a sinusoid with a frequency of 0.1 Hz and an amplitude of 1000 rad/s, which represents around 38% of the maximum angular speed of the motor. The sensor fault is also a sinusoid with a frequency of 0.25 Hz and an amplitude of 10 m/s. These faults are more aggressive than the ones considered in the previous cases, although the impact on the observer performance is small, as seen in the comparison between the system and observer outputs (Figure 10c).

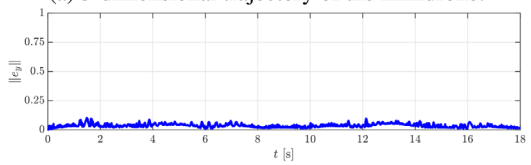
The switching function $e_y(t)$, shown in Figure 10b, shows that the observer is able to maintain sliding throughout the flight test, despite the presence of more aggressive faults. Figure 10d shows the fault reconstruction in this case. The observer tracks both sinusoidal faults, albeit with a small phase offset. This effect is more prominent in the actuator fault reconstruction due to its response being slower than the sensor fault reconstruction. Nonetheless, the phase offset is only 0.48 s, that is, a phase difference of only around 18°.

5 | CONCLUSION

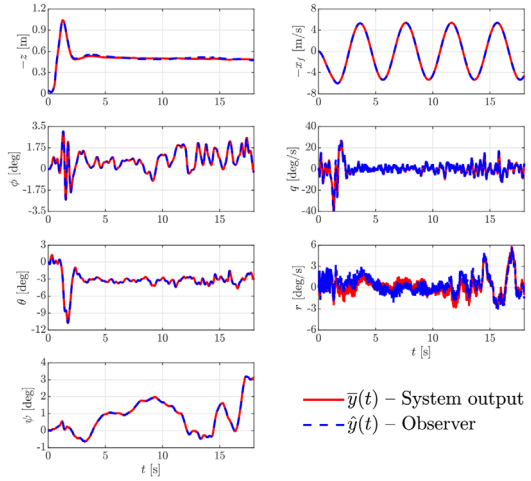
This paper presents the design and implementation of an LPV sliding mode observer for simultaneous actuator and sensor fault reconstruction in a Parrot® Rolling Spider minidrone. The observer is designed using an LPV model of the minidrone obtained from a 12 degree-of-freedom nonlinear model. The design procedure and numerical synthesis are explained and show how the observer can be obtained by solving a convex optimization problem that can be efficiently solved by commercially available software. The flight test results demonstrate that the sliding mode observer is able to accurately track the system's outputs, even in the presence of simultaneous actuator and sensor faults. The fault reconstruction is shown to work satisfactorily even in the presence of acute faults during flight. The successful implementation of this sliding mode FDD scheme within the limited computational processing capability available on the Rolling Spider minidrone reinforces the claim that this approach is numerically viable and powerful in embedded systems with limited computing power. Future work will be directed towards designing a fault-tolerant sliding mode controller that can take into account the fault reconstructions provided by the observer, thus improving the resilience of the



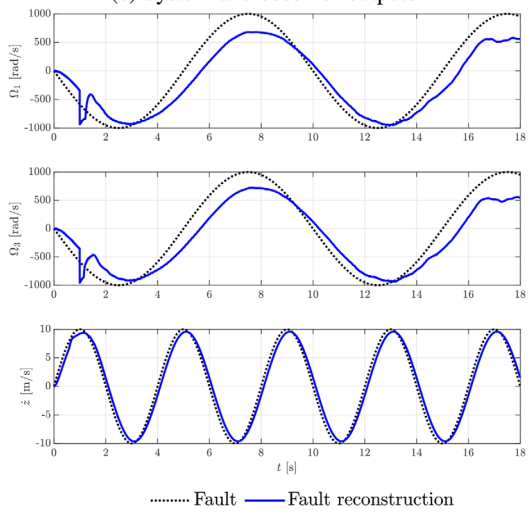
(a) 3-dimensional trajectory of the minidrone.



(b) Switching function.



(c) System and observer outputs.



(d) Fault reconstruction.

FIGURE 10 Case F4

minidrone to faults. The implementation of higher-order and super-twisting algorithms [42, 43] for fault detection in the minidrone is also envisaged, with the goal of reducing chattering.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

S. Waitman https://orcid.org/0000-0002-1127-3018
H. Alwi https://orcid.org/0000-0002-2892-6660
C. Edwards https://orcid.org/0000-0002-2222-0640

REFERENCES

1. de Miguel Molina, B., Segarra Oña, M.: The drone sector in Europe. In: SpringerBriefs in Law, pp. 7–33. Springer International Publishing, Switzerland (2017). https://doi.10.1007/978-3-319-71087-7_2
2. Wild, G., et al.: Exploring civil drone accidents and incidents to help prevent potential air disasters. Aerospace 3(3), 22 (2016). https://doi.10.3390/aerospace3030022
3. Edwards, C., et al. (eds.): Fault Tolerant Flight Control: A Benchmark Challenge. Lecture Notes in Control and Information Sciences, vol. 399. Springer, Berlin, Heidelberg, (2010). https://doi.10.1007/978-3-642-11690-2
4. Frangenberg, M., et al.: Fast actuator fault detection and reconfiguration for multicopters. In: AIAA Guidance, Navigation, and Control Conference. American Institute of Aeronautics and Astronautics, Washington DC (2015). https://doi.10.2514/6.2015-1766
5. Li, D., Liu, H. H.: Sensor bias fault detection and isolation in a large multirotor aerial vehicle using active disturbance rejection control. In: 2018 AIAA Information Systems-AIAA Infotech @ Aerospace. American Institute of Aeronautics and Astronautics, Washington DC (2018). https://doi.10.2514/6.2018-0249
6. Meskin, N., et al.: Fault detection and isolation strategy for a network of unmanned vehicles in presence of large environmental disturbances. In: AIAA Guidance, Navigation, and Control Conference. American Institute of Aeronautics and Astronautics, Washington DC (2009). https://doi.10.2514/6.2009-5656
7. Sadeghzadeh, I., Zhang, Y.: A review on fault-tolerant control for unmanned aerial vehicles (UAVs). In: Infotech@Aerospace. American Institute of Aeronautics and Astronautics, Washington DC (2011). https://doi.10.2514/6.2011-1472
8. Valenti, M., et al.: Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery. In: AIAA Guidance, Navigation, and Control Conference Exhibit. American Institute of Aeronautics and Astronautics, Washington DC (2006). https://doi.10.2514/6.2006-6200
9. Alwi, H., et al.: Fault Detection and Fault-Tolerant Control Using Sliding Modes. Springer, London, (2011). https://doi.10.1007/978-0-85729-650-4
10. Edwards, C., Spurgeon, S. K.: Sliding Mode Control: Theory and Applications. Series in Systems and Control. CRC Press, Boca Raton (1998)
11. Edwards, C., et al.: Sliding mode observers for fault detection and isolation. Automatica 36(4), 541–553 (2000). https://doi.10.1016/s0005-1098(99)00177-6
12. Edwards, C., Spurgeon, S. K.: A sliding mode control observer based FDI scheme for the ship benchmark. European J. Control 6, 341–356 (2000)
13. Zhang, H., Wang, J.: Active steering actuator fault detection for an automatically-steered electric ground vehicle. IEEE Trans. Veh. Technol. 66(5), 3685–3702 (2016). https://doi.10.1109/tvt.2016.2604759
14. Zhang, H., Wang, J.: Adaptive sliding-mode observer design for a selective catalytic reduction system of ground-vehicle diesel engines. IEEE/ASME Trans. Mechatronics 21(4), 2027–2038 (2016). https://doi.10.1109/tmech.2016.2542362

15. Tan, C. P., Edwards, C.: Sliding mode observers for robust detection and reconstruction of actuator and sensor faults. Int. J. Robust Nonlinear Control 13(5), 443–463 (2003). https://doi.10.1002/rnc.723
16. Alwi, H., Edwards, C.: Robust fault reconstruction for linear parameter varying systems using sliding mode observers. Int. J. Robust Nonlinear Control 24(14), 1947–1968 (2013). https://doi.10.1002/rnc.3009
17. Alwi, H., et al.: Reconstruction of simultaneous actuator and sensor faults for the RECONFIGURE benchmark using a sliding mode observer. IFAC Proc. Volumes 47(3), 3497–3502 (2014). https://doi.10.3182/20140824-6-za-1003.02175
18. Ben Brahim, A., et al.: Simultaneous actuator and sensor faults reconstruction based on robust sliding mode observer for a class of nonlinear systems. Asian J. Control 19(1), 362–371 (2016). https://doi.10.1002/asjc.1359
19. Tan, J., et al.: Sliding mode fault tolerant control for unmanned aerial vehicle with sensor and actuator faults. Sensors 19(3), 643 (2019). https://doi.10.3390/s19030643
20. Khosravi, A., et al.: A neuro-fuzzy based sensor and actuator fault estimation scheme for unknown nonlinear systems. In IEEE Int. Joint Conf. Neural Netw. IEEE, (2005). https://doi.10.1109/ijcnn.2005.1556266
21. Li, X., Zhu, F.: Simultaneous actuator and sensor fault estimation for descriptor LPV system based on H_∞ reduced-order observer. Optimal Control Appl. Methods 37(6), 1122–1138 (2015). https://doi.10.1002/oca.2226
22. Li, X., Zhu, F.: Simultaneous time-varying actuator and sensor fault reconstruction based on PI observer for LPV systems. Int. J. Adaptive Control Signal Process. 29(9), 1086–1098 (2015). https://doi.10.1002/acs.2522
23. Chandra, K. P. B., et al.: Fault reconstruction for a quadrotor using an LPV sliding mode observer. IFAC-PapersOnLine 48(21), 374–379 (2015). https://doi.10.1016/j.ifacol.2015.09.555
24. Li, T., et al.: Passive and active nonlinear fault-tolerant control of a quadrotor unmanned aerial vehicle based on the sliding mode control technique. Proc. Inst. Mech. Engineers, Part I: J. Syst. Control Eng. 227(1), 12–23 (2012). https://doi.10.1177/0959651812455293
25. Nguyen, N. P., Hong, S. K.: Sliding mode thau observer for actuator fault diagnosis of quadcopter UAVs. Appl. Sci. 8(10), 1893 (2018). https://doi.10.3390/app8101893
26. Parrot, S.A.: Parrot Minidrones Rolling Spider: User Guide (2014)
27. MathWorks.: Simulink® Support Package for Parrot® Minidrones: User's Guide, (2019)
28. Karaman, S., Riether, F.: Getting started with MIT's Rolling Spider MATLAB Toolbox. Massachusetts Institute of Technology, (2016)
29. Riether, F.: Agile quadrotor maneuvering using tensor-decomposition-based globally optimal control and onboard visual-inertial estimation. Master's thesis, Massachusetts Institute of Technology, (2016)
30. Bouabdallah, S.: Design and control of quadrotors with application to autonomous flying. PhD dissertation, Ecole Polytechnique Fédérale de Lausanne, (2007)
31. Alwi, H., Edwards, C.: Sliding mode fault-tolerant control of an octorotor using linear-parameter-varying-based schemes. IET Control Theory Appl. 9(4), 618–636 (2015). https://doi.10.1049/iet-cta.2014.0215
32. Utkin, V. I.: Sliding Modes in Control and Optimization. Springer, Berlin Heidelberg, (1992). http://doi.10.1007/978-3-642-84379-2
33. Boyd, S., et al.: Linear Matrix Inequalities in System & Control Theory. Studies in Applied Mathematics, vol. 15. Society for Industrial & Applied Mathematics, Philadelphia (1994)
34. Alwi, H., Edwards, C.: Fault detection and fault-tolerant control of a civil air-craft using a sliding-mode-based scheme. IEEE Trans. Control Syst. Technol. 16(3), 499–510 (2008). http://doi.10.1109/tcst.2007.906311
35. Löfberg, J.: YALMIP : a toolbox for modeling and optimization in MATLAB. In: International Conference on Robotics and Automattion, pp. 284–289. IEEE, Piscataway (2004). http://doi.10.1109/cacsd.2004.1393890
36. MOSEK ApS.: The MOSEK optimization toolbox for MATLAB manual. Version 9.0. (2019)
37. Tóth, R., et al.: Discretisation of linear parameter-varying state-space representations. IET Control Theory & Appl. 4(10), 2082–2096 (2010). http://doi.10.1049/iet-cta.2009.0572
38. Basford, P. J., et al.: Performance analysis of single board computer clusters. Future Gener. Comput. Syst. 102, 278–291 (2020). http://doi.10.1016/j.future.2019.07.040
39. Nikolskiy, V., Stegailov, V.: Floating-point performance of ARM cores and their efficiency in classical molecular dynamics. J. Phys.: Conf. Ser. 681, 012049 (2016). http://doi.10.1088/1742-6596/681/1/012049
40. Intel: APP metrics for Intel®microprocessors. https://www.intel.com/content/dam/support/us/en/documents/processors/APP-for-Intel-Core-Processors.pdf. Accessed 27 November 2020.
41. Qian, H.: Counting the floating point operations (FLOPS). MATLAB Central File Exchange (2020) https://www.mathworks.com/matlabcentral/fileexchange/50608-counting-the-floating-point-operations-flops. Accessed 11 November 2020.
42. Mellucci, C., et al.: Second-order sliding mode observers for fault reconstruction in power networks. IET Control Theory Appl. 11(16), 2772–2782 (2017). http://doi.10.1049/iet-cta.2017.0249
43. Nagesh, I., Edwards, C.: A multivariable super-twisting sliding mode approach. Automatica 50(3), 984–988 (2014). http://doi.10.1016/j.automatica.2013.12.032

How to cite this article: Waitman, S., Alwi, H., Edwards, C. Flight evaluation of simultaneous actuator/sensor fault reconstruction on a quadrotor minidrone. IET Control Theory Appl. 1–16 (2021). https://doi.org/10.1049/cth2.12180