Journal of Intelligent & Robotic Systems (2024) 110:148
https://doi.org/10.1007/s10846-024-02176-2

SHORT PAPER

Motor Fault Detection and Isolation for Multi-Rotor UAVs Based on External Wrench Estimation and Recurrent Deep Neural Network

Jonathan Cacace[1] · Vincenzo Scognamiglio[2] · Fabio Ruggiero[2] · Vincenzo Lippiello[2]

Received: 11 May 2023 / Accepted: 13 September 2024

Abstract

Fast detection of motor failures is crucial for multi-rotor unmanned aerial vehicle (UAV) safety. It is well established in the literature that UAVs can adopt fault-tolerant control strategies to fly even when losing one or more rotors. We present a motor fault detection and isolation (FDI) method for multi-rotor UAVs based on an external wrench estimator and a recurrent neural network composed of long short-term memory nodes. The proposed approach considers the partial or total motor fault as an external disturbance acting on the UAV. Hence, the devised external wrench estimator trains the network to promptly understand whether the estimated wrench comes from a motor fault (also identifying the motor) or from unmodelled dynamics or external effects (i.e., wind, contacts, etc.). Training and testing have been performed in a simulation environment endowed with a physic engine, considering different UAV models operating under unknown external disturbances and unexpected motor faults. To further assess this approach's effectiveness, we compare our method's performance with a classical model-based technique. The collected results demonstrate the effectiveness of the proposed FDI approach.

Keywords Fault detection and isolation · Long short-term memory networks · External wrench estimation · Safe aerial robotics

1 Introduction

In the last decade, the use of aerial vehicles to perform service tasks is widely increased. Among the different types, vertical take-off and landing (VTOL) unmanned aerial vehicles (UAVs) are suitable for performing several tasks like inspection, surveillance, search and rescues, thanks to their agility, fast motion capabilities, and their ability to hover during the flight. In this context, e-shopping companies [1] are planning to use UAVs for home delivery of commercial goods, while different applications require aerial vehicles in safety-critical environments. The latter is the inspection of oil and gas facilities, in which drones can perform visual and contact inspection of pipelines transporting fluid [2, 3]. Safety remains a significant concern in these domains to prevent hurts to human operators or dangerous equipment damages.

The actuation system of a multi-copter is composed of different brushless motors with fixed or actuated propellers. Eventual damages to the UAV's propellers cause a significant decrease in its propulsion system, compromising stable flying capabilities. Hence, fault detection is an essential feature to implement multi-rotor safety [4], representing this work's primary motivation. Motor fault detection is a complex task since UAV rotors usually do not provide any feedback. For this reason, typical failure detection techniques use onboard sensor measurements.

This work exploits a deep recurrent neural network fed by estimated disturbances to detect and isolate possible motor faults. During the flight, the UAV is subject to many unmodelled aerodynamic disturbances affecting its behavior and stability [5]. These disturbances are often seen as the effect

✉ Vincenzo Scognamiglio
vincenzo.scognamiglio2@unina.it

Jonathan Cacace
jonathan.cacacae@eurecat.org

Fabio Ruggiero
fabio.ruggiero@unina.it

Vincenzo Lippiello
vincenzo.lippiello@unina.it

[1] Eurecat, Centre Tecnologic de Catalunya Robotics and Automation Unit, Parc Tecnològic del Vallès, Av. Universitat Autònoma, 23, Cerdanyola del Vallès 08290, Spain

[2] Department of Electrical Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, Naples 80125, Italy

Published online: 08 October 2024

of a lumped external wrench at the UAV's center of mass. Suitable estimators are designed to improve the tracking performance of the onboard controllers [5]. The idea behind this work is that if a UAV's motor is partially or entirely broken, the effect can be seen as another disturbance preventing the satisfactory performance of the tracking controller. The available estimators in the literature cannot distinguish between the single external effects creating the lumped external wrench at the UAV's centre of mass. Therefore, an assistant system must come with the estimator to understand whether the estimated disturbance is caused or not by a partial or entire failure of a motor of the UAV. Here, the assisting system is a recurrent neural network (RNN) [6] composed of long short-term memory (LSTM) [7] nodes analyzing the estimated values. The network is trained to detect and isolate the fault when the UAV flies random trajectories subject to external disturbances (i.e., wind), including randomly generated motor faults. The proposed FDI approach has been carried out in the Gazebo simulator, a widely used robot simulator endowed with a physic engine. To simulate the dynamics of the UAV and the behavior of its motors, the RotorS ROS package [8] has been used. This package allows simulating of the multi-copter dynamics and its propellers. Three different UAV frames have been considered: two quadcopters, one adopting the *plus* configuration and one flying with the *cross* design, and a hexacopter. These configurations are depicted in Fig. 1. In particular, Fig. 1(a) describes the plus configuration in which a single rotor leads the platform, while Figs. 1(b) and (c) report a cross configuration, in which two rotors lead the aircraft motion.

The rest of the manuscript is organized as follows. Section 2 presents an overview of multi-copter fault detection methods and the proposed contributions against the current state of the art. Section 3 presents the system implementation, describing the UAV controller, the unmodeled disturbances estimator, and the LSTM structure. The use of this network is discussed in Section 4. Finally, the simulation case studies are presented in Section 5, and the obtained results are discussed therein. The conclusion and future work are finally illustrated.

2 Related Works and Contributions

Autonomous UAVs are complex systems. They have many essential sensors for safe and reliable stabilization and navigation. The onboard avionic comprises an inertia measurement unit (IMU), a flight control unit, and multiple motors. If one of these elements is subjected to faults or damages, the capacity of the aerial platform to successfully fly and land without crashes is compromised. Two primary sources of faults can happen in the system flight: sensor faults [9–11] and actuator's fault. For this reason, many UAVs include redundancy in their actuation or electronics [12, 13], while in [14] authors present a novel open-source design for hexacopters robust to the failure of any motor/propeller.

Different fault-tolerant controller methods have been proposed to allow UAVs to fly when a motor is corrupted or completely lost. Some of these methods rely on motor redundancy [15], while other approaches, like in this paper, consider custom controllers achieving autonomous and stable flight even in the case of not-redundant multi-copters (i.e. quadcopters) [16, 17]. Therefore, these methods typically need to detect the presence of a rotor fault quickly and, in some cases, to identify which motor is not working as expected [18].

In this work, we consider the possibility of identifying faults in the UAV rotors. The main complication in this task is that, typically, there are no onboard sensors to retrieve a rotor's output (and the status). Hence, FDI techniques rely on state estimators that can be used to assess the effectiveness of a rotor. However, authors in [19] used current sensors and the onboard accelerometer to detect motor faults. In contrast, in [20], external sensors (i.e., audio input sensors) have been exploited to classify propeller corruptions by measuring the noise emitted by the UAV during the flight. Unlike these contributions, the presented method here does not require additional sensors. Other state-of-the-art solutions following the same principles (i.e., directly exploiting onboard sensor measurements to characterize the fault,) rely on a classical Luenberger linear estimator [21] or a Thau observer [22, 23]. A nonlinear adaptive estimator is instead implemented in [24], using a bank of nonlinear adaptive fault isolation estimators to identify which rotor has the fault. In [25, 26], a two-stage Kalman filter is used, while in [27], an extended Kalman filter is introduced to monitor the health of each motor. The IMU sensor represents the principal information source of these FDI methods. Similarly, a multiple integral fault detection filter (PMI) is proposed in [28] to estimate sensor noise and detect system faults. Other sensor and actuator fault diagnosis methods are collected in [29].

The proposed approach estimates unmodelled disturbances acting on the UAV frame using a momentum-based external wrench estimator [30]. This estimator has already been used on quadrotors to enhance its position control loop, compensating for detected disturbance effects [31, 32]. This work only uses this generalized external forces estimator to characterize rotor power loss. However, this estimator can only provide the lumped effect wrench at the centre of mass resulting from the many unmodelled effects acting on the UAV without distinguishing between them. To this aim, a data-driven approach is employed to understand whether the estimated disturbance is caused or not by a partial or entire failure of a motor of the UAV. Data-driven and machine-learning approaches have also been used standalone to solve actuator FDI problems on UAVs [33–35]. Similarly, we pro-

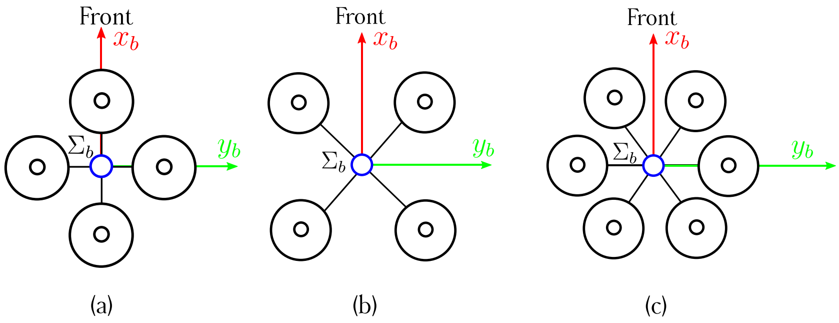


Fig. 1 Different frame configurations. *(a)* - quadcopter plus configuration *(b)* - quadctoper cross configuration, *(c)* - hexacopter cross configuration

posed a deep recurrent neural network to identify eventual faults. Deep learning techniques have been widely used to solve many problems in different scenarios. Authors in [36] deployed an LSTM network to predict UAV malfunctions by analyzing the vibration of the aerial platform. Authors in [37] use a deep learning approach to identify causes of failure after crashes or incidents. Again, authors in [38] used a data-driven approach based on LSTM networks to detect onboard sensor drift. Instead, we estimate the presence of external forces like a loss of thrust from one of the motors to identify a fault correctly.

To summarize, the provided contributions against state-of-the-art approaches are (i) identifying the rotor fault as an external disturbance on the UAV and estimating the lumped wrench on its center of mass; (ii) deploy a data-driven system able to understand whether the estimated wrench is generated by a (partial or entire) rotor fault and identify the fault source; (iii) conduct a simulation campaign comparing the obtained results with a model-based approach available in the literature.

3 System Architecture

The proposed system architecture is depicted in Fig. 2. It is composed of the following elements: (i) a *Geometric Controller* for the UAV that generates the force and control moments for the aerial platform later translated into propeller's velocities; (ii) an *External Wrench Estimator* to calculate the unmodelled disturbances acting on the platform; (iii) a Long Short-time Memory neural network module to detect and isolate motor faults. Besides, a simple *Trajectory Planner* streams the desired position and orientation of the UAV. At the same time, we assume that the UAV can estimate its position and attitude in a fixed inertial frame. Before detailing the architecture modules, we briefly introduce the model of standard multi-rotor UAVs.

3.1 Dynamic Model of Multi-rotors

Flat multi-rotors are under-actuated systems having six degrees of freedom in the considered mathematical model but only four control inputs. Let Σ_w and Σ_b be two frames representing the world fixed and the body-fixed frame attached to the UAV centre of mass, respectively (see Fig. 1). The position of Σ_b in the world fixed frame is denoted by $\mathbf{p}_b = [\, x \;\; y \;\; z \,]^T \in \mathbb{R}^3$ whereas its attitude is described by the rotation matrix $\mathbb{R}_b \in SO(3)$. In this context, the dynamic equations of the UAV are described as follows [32].

$$m\ddot{\mathbf{p}}_b = mg\mathbf{e}_3 - u_T\mathbf{R}_b\mathbf{e}_3 + \mathbf{f}_{\text{ext}} \tag{1a}$$

$$\dot{\mathbf{R}}_b = \mathbf{R}_b S(\boldsymbol{\omega}_b^b) \tag{1b}$$

$$\mathbf{I}_b\dot{\boldsymbol{\omega}}_b^b = -S(\boldsymbol{\omega}_b^b)\mathbf{I}_b\boldsymbol{\omega}_b^b + \boldsymbol{\tau}^b + \boldsymbol{\tau}_{\text{ext}}^b, \tag{1c}$$

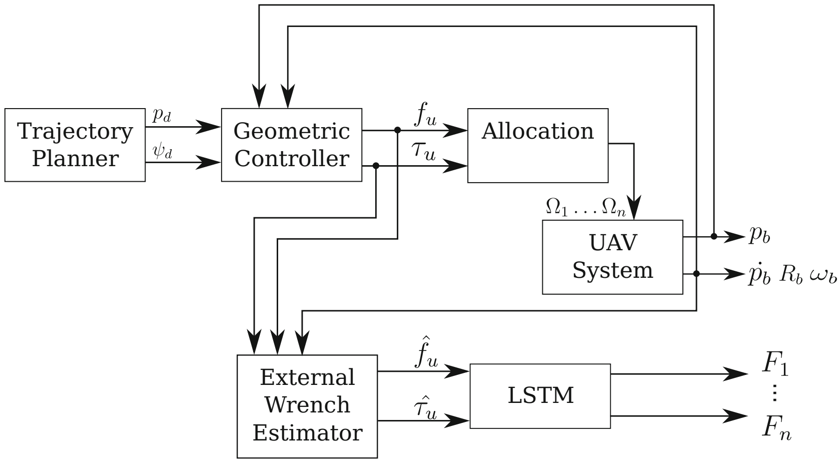
where $\mathbf{e}_3 = [0 \;\; 0 \;\; 1]^T \in \mathbb{R}^3$, $m > 0$ is the UAV mass, $\mathbf{I}_b \in \mathbb{R}^{3\times 3}$ is its symmetric and positive definite inertia matrix expressed in Σ_b, $\boldsymbol{\omega}_b^b \in \mathbb{R}^3$ is the angular velocity vector expressed in Σ_b, $S(\cdot) \in \mathbb{R}^{3\times 3}$ is the skew-symmetric operator, $u_T \in \mathbb{R}$ and $\boldsymbol{\tau}^b = [\, \tau_x \;\; \tau_y \;\; \tau_z \,]^T \in \mathbb{R}^3$ are the total thrust force and the control torques, respectively, and $\mathbf{f}_{\text{ext}} \in \mathbb{R}^3$ and $\boldsymbol{\tau}_{\text{ext}}^b \in \mathbb{R}^3$ the lumped vectors denoting unknown forces and torques (this last expressed in Σ_b), respectively, acting on the vehicle (e.g., aerodynamic and buoyancy effects, flapping dynamics, parametric uncertainties, imbalances caused by batteries and/or on-board sensors, wind gusts, interaction with the environment, propellers faults, etc.).

The desired force and torque must be translated into rotational velocities for the multi-rotor propellers to actuate the aerial platform. This step depends on the adopted frame and, more precisely, on the number of motors and their position on the frame. As already stated, this work considers three different UAV models and the general form of this transformation is detailed in the following.

Considering a UAV with n propellers, let $\Omega_i \in \mathbb{R}$, with $i = 1, \ldots, n$, be the rotational velocities of the propellers, the multicopter's control input $\mathbf{u} = [\, u_T \;\; \boldsymbol{\tau}^{b^T} \,]^T$ can be computed as follows (See [39])

$$\mathbf{u} = \begin{bmatrix} c_t & c_t & \ldots & c_t \\ \sin(\alpha_1)\, l_1\, c_t & \sin(\alpha_2)\, l_2\, c_t & \ldots & \sin(\alpha_n)\, l_n\, c_t \\ \cos(\alpha_1)\, l_1\, c_t & \cos(\alpha_2)\, l_2\, c_t & \ldots & \cos(\alpha_n)\, l_n\, c_t \\ d\, c_a & c_a & \ldots & c_a \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \ldots \\ \Omega_n^2 \end{bmatrix}, \tag{2}$$

Fig. 2 System architecture



where $\alpha_i \in \mathbb{R}$ is the angle of the rotor on the UAV frame with respect to Σ_b, $l_i > 0$ is the distance between the $i-th$ propeller and the origin of Σ_b, $c_t > 0$, $c_a > 0$ are the thrust and torque coefficients of the propellers, respectively, and d is the rotation direction of the motor: positive/negative for clockwise/counterclockwise rotor rotations.

3.2 Geometric Controller

In this section, the *Geometric Controller* module of the system architecture is detailed. This module aims to receive the desired position, velocity, and orientation of the UAV from the *Trajectory Planner* to generate the desired thrust and torques (u_T and $\boldsymbol{\tau}^b$) to apply to the UAV system. In particular, this module receives the desired position and velocity of the Σ_b's origin in Σ_w $\mathbf{p}_d$, $\dot{\mathbf{p}}_d \in \mathbb{R}^3$ and the desired orientation around the z-axis of the body frame (ψ_d, namely the desired yaw angle) with respect to the fixed world frame. Considering the under-actuation of the system, a hierarchical approach has been considered to control both the position, $\mathbf{p}_b$ (outer loop), and the attitude (inner loop), $\mathbf{R}_b$, of the multi-rotor. In this context, we adopt the geometric tracking controller in $SE(3)$ proposed in [40]. The outer position loop tracking errors are

$$\begin{aligned} \mathbf{e}_p &= \mathbf{p}_b - \mathbf{p}_d, \\ \mathbf{e}_v &= \dot{\mathbf{p}}_b - \dot{\mathbf{p}}_d, \end{aligned} \tag{3}$$

Let $\mathbf{R}_d \in SO(3)$ be the desired rotation matrix specifying the desired orientation of the UAV and $\mathbf{x}_d \in \mathbb{R}^3$ be the axis from ψ_d. The necessary thrust u_T and the desired body axis $\mathbf{z}_d \in \mathbb{R}^3$ can be computed as follow.

$$u_T = (\mathbf{K}_p\,\mathbf{e}_p + \mathbf{K}_v \mathbf{e}_v + mg\mathbf{e}_3 - m\ddot{\mathbf{p}}_d)^T \mathbf{R}_b \mathbf{e}_3 \tag{4a}$$

$$\mathbf{z}_d = -\frac{-\mathbf{K}_p \mathbf{e}_p - \mathbf{K}_v \mathbf{e}_v - mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_{b,d}}{\| -\mathbf{K}_p \mathbf{e}_p - \mathbf{K}_v \mathbf{e}_v - mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_d \|}, \tag{4b}$$

where $\mathbf{K}_p \in \mathbb{R}^{3\times3}$ and $\mathbf{K}_v \in \mathbb{R}^{3\times3}$ are positive definite gain matrices, while $\|\cdot\| \in \mathbb{R}^3$ is the Cartesian norm. We can now obtain the desired rotation matrix through

$$\mathbf{R}_d = \left[S\left(\frac{S(\mathbf{z}_d)\mathbf{x}_d}{\|S(\mathbf{z}_d)\mathbf{x}_d\|} \right) \mathbf{z}_d \;\; \frac{S(\mathbf{z}_d)\mathbf{x}_d}{\|S(\mathbf{z}_d)\mathbf{x}_d\|} \;\; \mathbf{z}_d \right]. \tag{5}$$

Now, we can define the tracking error of the inner loop of the controller as follows

$$\mathbf{e}_R = \frac{1}{2}\,(\mathbf{R}_{b,d}^T\,\mathbf{R}_b - \mathbf{R}_b^T\,\mathbf{R}_{b,d})^\vee, \tag{6a}$$

$$\mathbf{e}_\omega = \boldsymbol{\omega}_b^b - \mathbf{R}_b^T\,\mathbf{R}_d\,\boldsymbol{\omega}_d^d, \tag{6b}$$

in which $\boldsymbol{\omega}_d^d \in \mathbb{R}^3$ is the desired body rotation velocity expressed in Σ_b and $^\vee : \mathbb{R}^{3\times3} \to \mathbb{R}^3$ is a map function performing the inverse of the skew-symmetric operator. Finally, the control torque is computed as

$$\begin{aligned} \boldsymbol{\tau}^b = &-\mathbf{K}_R\,\mathbf{e}_R - \mathbf{K}_\omega\,\mathbf{e}_\omega + S(\boldsymbol{\omega}_b^b)\,\mathbf{I}_b\,\boldsymbol{\omega}_b^b + \\ &- \mathbf{I}_b\,[S(\boldsymbol{\omega}_b^b)\,\mathbf{R}_b^T\,\mathbf{R}_d\,\boldsymbol{\omega}_d^d - \mathbf{R}_b^T\,\mathbf{R}_d\,\dot{\boldsymbol{\omega}}_d^d], \end{aligned} \tag{7}$$

where $\mathbf{K}_R \in \mathbb{R}^{3\times3}$ and $\mathbf{K}_\omega \in \mathbb{R}^{3\times3}$ are positive definite gain matrices,

3.3 External Wrench Estimator

The goal of the *External Wrench Estimator* module is to estimate the unmodeled disturbances acting on the UAV frame. It implements the momentum-based estimator already used in [31, 41] to improve the position controller of a quadrotor. The input of this module is the desired control force and torque (f_u, τ_u), the estimated linear and angular velocity, and orientation in the fixed body frame. These disturbances are characterized as generalized external forces acting on the robot's body. In this context, the forces $\hat{\mathbf{F}}_{\text{ext}}(t) = \left[\hat{\mathbf{f}}_{\text{ext}}^T(t)\ \hat{\boldsymbol{\tau}}_{\text{ext}}^{b^T}(t)\right]^T$ are calculated as follows

$$\hat{\mathbf{F}}_{\text{ext}}(t) = \mathbf{K}_1 \Bigg(\int_0^t -\hat{\mathbf{F}}_{\text{ext}}(\sigma) + \mathbf{K}_2 \Bigg(\alpha(\sigma) - \int_0^t \left(\begin{bmatrix} u_T \mathbf{R}_b \mathbf{e}_3 - mg\mathbf{e}_3 \\ \tau^b - \mathbf{S}(\omega_b^b)\mathbf{I}_b\omega_b^b \end{bmatrix} + \hat{\mathbf{F}}_{\text{ext}}(\sigma) \right) \mathrm{d}\sigma \Bigg) \mathrm{d}\sigma \Bigg) \tag{8}$$

where the matrices $\mathbf{K}_1 \in \mathbb{R}^{6\times6}$ and $\mathbf{K}_2 \in \mathbb{R}^{6\times6}$ are positive-definite gains, while $\boldsymbol{\alpha} \in \mathbf{R}^6$ is the system's momentum

$$\boldsymbol{\alpha} = \begin{bmatrix} m\mathbf{I}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{I}_b \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_b \\ \boldsymbol{\omega}_b^b \end{bmatrix}. \tag{9}$$

One of the novelties of this work lies in using the output of the external wrench estimator to train an assistant system, which is a neural network. This neural network leverages the estimation of forces and torques, combined with the knowledge of whether a fault was injected into a rotor. In subsequent testing phases, the torques and forces from the estimator serve as inputs to the trained network, enabling it to distinguish whether the external forces acting on the UAV are due to an external disturbance (e.g., wind) or a rotor fault.

4 Rotor Fault Detection and Isolation

In this section, the fault detection and isolation process is detailed. We propose an RNN based on LSTM [42] to detect motor faults. This structure has been introduced to address the vanishing gradient problem of the RNNs [43]. In particular, the hidden unit of a traditional RNN is replaced by a memory cell to handle the information received as input by the network. This is made exploiting three *gates*: the input gate, forget gate, and output gate that act as regulators for the manipulation and the utilization of the memory discerning between relevant and irrelevant information; in this context, LSTM networks are particularly suited to analyze time-series data.

The input layer is composed of six nodes, representing the three-dimensional forces and torques estimated by the *External Wrench Estimator* module (i.e., $\hat{\mathbf{f}}_{\text{ext}}$, $\hat{\tau}_{\text{ext}}$). In contrast, the output layer's size depends on the UAV's number of motors. Each output node evaluates the operating status of a motor of the aerial platform. The structure of the proposed neural network is depicted in Fig. 3. It consists of an input layer, a hidden layer of LSTM cells, and an output layer associated with a *softmax* function. We deployed two different networks, one with 4 output nodes to detect rotor faults on the two quadcopter models and one with 6 output nodes used with the hexacopter. Besides, a hidden layer composed of 25 nodes has been considered for both models. To train the network, we considered 100 epochs for each batch of data, while the size of a batch has been experimentally set to 20 samples. Finally, the *Adam* optimizer and the *categorical cross-entropy* loss function has been used.

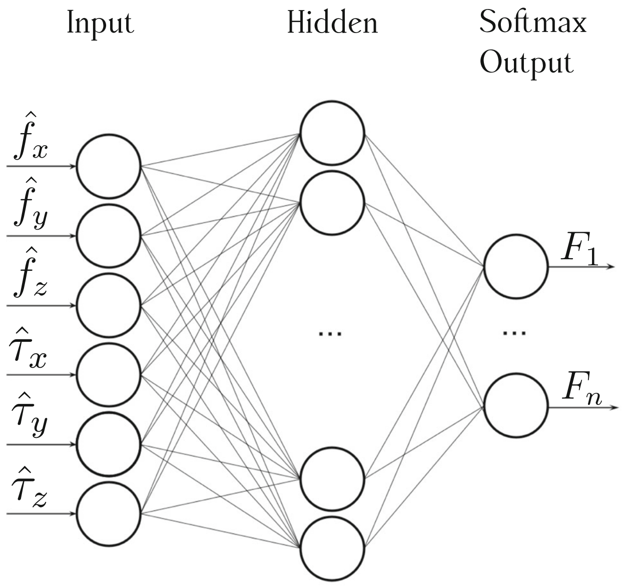


Fig. 3 LSTM Neural network structure

System Training

In the proposed application, the estimated unmodeled disturbances in the presence of a motor fault strictly depend on the system dynamics and the configuration of the rotors, the deployed LSTM network must be trained for each platform. For this reason, three different datasets have been created simulating several faulty trajectories considering a set of static and random parameters. The complete list of the parameters with their boundaries values are reported in Table 1. In particular, a new trajectory is planned for a platform with r motors, considering random target way-points ($[x, y, z]$ and ψ) and cruise velocity (cv). The target is generated using random functions from standard C++ libraries based on a uniform distribution.

Table 1 Different training parameters using to generate the datasets

Parameter	Values
r	4,6
x, y	$[-10; 10]\ m$
z	$[0.5; 12]\ m$
ψ	$[0; 2\pi]\ rad$
cv	$[0.1; 2.0]\ m/s$
m	$[1; r]$
g_{th}	$[0; 500]$
f_k	$[0.01; 1.0]$

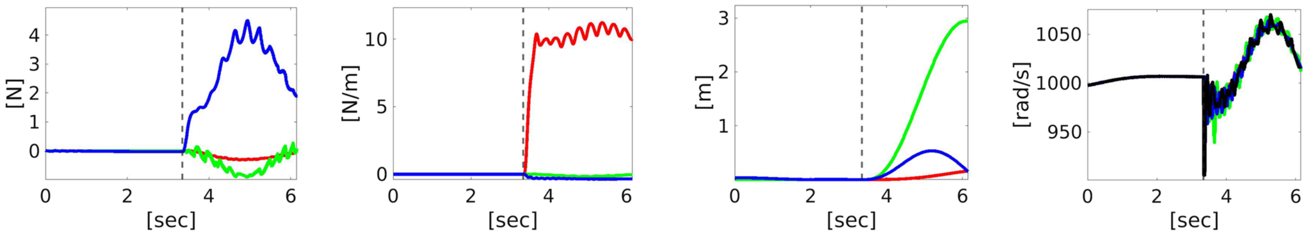


Fig. 4 Normal condition (before dashed line) and faulty condition (after dashed line). From left to right graphics represent the estimated unmodeled forces, estimated unmodeled torques, position error and motor velocities

The planned waypoints are limited between 0.5 and 12 m. The lower limit is set to avoid generating trajectories too close to the ground, while the upper limit defines medium-length flights. However, this upper limit can be adjusted to perform trajectories at different altitudes as needed. At the same time, during the flight, we randomly decide to inject a fault into one of the motors of the aerial platform. When a generated number (g_{th}) is higher than a certain threshold (e.g., 500 in the proposed training sessions), a new fault is provoked. This number has been experimentally selected to allow a balanced time of flight with and without motor fault. Let ω_i be the velocity of a UAV propeller, $f_k\,\omega_i$ simulates the loss of power of motor m. Like target way-points, f_k and m are randomly selected during the navigation.

During each training session, we recorded the disturbances and the motor status vector consisting of n binary values: $\mathbf{f} = [f_1, \ldots, f_n]$. In this context, n is the number of the aircraft's motor, and f_i equals one if a fault is injected on the $i-th$ motor, zero otherwise. The estimation process runs at 100 Hz, and the same framerate stores data in the dataset. It is worth noticing that we did not consider more than one rotor fault for time.

An example of the estimated unmodeled disturbances with and without a rotor fault is shown in Fig. 4, where the system force, torque, position error, and motor velocities are depicted. In this picture, the dashed line indicates that a fault on one of the motors of the UAV has been injected. In this case, that motor is commanded to lose 10% of its efficiency.

Finally, a training session ends when a certain amount of time elapses after a motor fault has been injected or the fault is too critical (e.g. the motor completely loses the spinning force) to compromise any platform stabilisation. Such a latter condition is recognised considering the attitude error of the onboard controller. When this error exceeds its control saturation (e.g., 0.35 rad), the system is reset, and a new training session starts.

Table 2 Precision, Recall and Accuracy indexes for each UAV platform

Platform	P	R	Acc
Quad (+)	0.98	0.92	0.97
Quad (×)	0.95	0.81	0.92
Hexa	0.98	0.73	0.88

The collected datasets have been sequentially split into training and test sets, covering the samples' 70% and 30%. Test results on the test datasets are reported in Table 2 in which the Precision (P), Recall (R) and the Accuracy (Acc) indexes are reported. Considering the classical definition of *true/false* positive as the correct/wrong identification of a motor fault (TP/FP), and *true/false* negative as the correct/wrong classification of normal working conditions of the UAV motors (TN/FN), these quantities are calculated as follows

- Precision (P): represents the ratio between the correct predictions and the total predictions:

$$P = \frac{TP}{TP + FP}$$

- Recall (R): represents the ratio of the correct predictions and the total number of correct items in the set:

$$R = \frac{TP}{TP + FN}$$

- Accuracy (Acc): represents the correct predicted data over all the elements of the dataset:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

Table 3 Confusion matrix for the quadcopter flying with the plus configuration

Quad (+)	F_1	F_2	F_3	F_4
F_1	0.93	0	0	0.0
F_2	0	0.9	0	0.0
F_3	0	0	0.92	0.0
F_4	0	0	0	0.94

Table 4 Confusion matrix for the quadcopter flying with the cross configuration

Quad (×)	F_1	F_2	F_3	F_4
F_1	0.7	0	0	0.06
F_2	0	0.77	0.01	0.0
F_3	0	0.03	0.84	0.0
F_4	0.06	0.006	0	0.85

The closer the Precision, Recall, and Accuracy values are to one, the higher the performance of the FDI method is good. Finally, in Tables 3, 4, and 5 the confusion matrices are reported. These results demonstrate that the deployed LSTM network can correctly isolate the fault and identify the damaged rotor with good accuracy. These results show that the system works better when the robot motion is not coupled on multiple rotors (see Table 3).

Besides, these results are obtained considering the samples stored in the datasets. In a more realistic scenario, false-negative phenomenons can be mitigated by considering over-time testing sessions, as validated in the next section.

5 Case Studies

This section presents the system evaluation through several simulated case studies meant to demonstrate the effectiveness of the proposed FDI method. Nowadays, several tools exist to simulate aerial robot dynamics (see [44, 45]). In this work, we rely on RotorS [8], a ROS-based simulator that provides a modular framework to design Micro Aerial Vehicles to test control and state estimation algorithms.

Figure 5 describes the software architecture implemented to perform the evaluation. As stated, we tested three UAV models using the Gazebo simulation environment. Tests were performed on a standard computer running Ubuntu 20.04 GNU/Linux OS and ROS Noetic as robotic middleware. The LSTM network has been implemented using TensorFlow library[1] through Keras[2] high-level interface programmed in Python language. As for the UAV models, they are inspired by the *Firefly* and *Hummingbird* platforms from *Ascending Technologies* and the *Iris* quadcopter from *3D Robotics* (see Fig. 6). The dynamic parameters of the simulated UAVs are reported in Table 6. Similarly, the controller gains used to test the system are reported in Table 7. These gains have to be multiplied by an identity matrix of dimension three, i.e., $k_p\mathbf{I}_3$.

Table 5 Confusion matrix for the hexacopter

Hexa	F_1	F_2	F_3	F_4	F_5	F_6
F_1	0.89	0.02	0	0	0	0
F_2	0	0.93	0	0	0	0
F_3	0	0	0.93	0	0	0
F_4	0	0	0	0.87	0	0.05
F_5	0	0	0	0	0.9	0
F_6	0.13	0	0	0	0	0.88

The multi-rotor is commanded to take off at a fixed altitude in the simulation case studies. Then, similarly to the training sessions, a set of waypoints is randomly generated to perform different trajectories. During this motion, rotor faults can take place at any moment. A new test is started by the *Session Manager* whose aim is to set up a new trajectory (i.e. x, y, z, ψ, cv). The *detector* module loads the LSTM-trained model based on the UAV under test and continuously evaluates the state of the multi-copter rotors. When the *detector* module reveals a new fault, it is compared with the output of the *Session Manager*. In this way, we can characterize correct or wrong classification results. After that, a new fault has been injected the simulation scene is reset to start a new testing session.

Two scenarios have been considered with and without external disturbances to demonstrate the detection system's effectiveness. In the first scenario, random trajectories were performed replicating the conditions of the *LSTM* network training phase. In the second scenario, external forces are generated to affect the flying platform with turbulence noise. In this context, these disturbances emulate wind gusts that represent one of the significant external disturbance sources of outdoor operating UAVs [46, 47]. During each test, we collected the following data.

- *Length*: the length of the total executed paths.
- *Time*: the total elapsed time during the tests.
- *Accuracy*: the true positive rate.
- *Detection time*: the elapsed time between the injection of a new fault and its detection.

Actuator Faults Without External Disturbances

Without loss of generality, consider the case of the quadrotor. Given $\hat{\mathbf{F}}_{\text{ext}}$, i.e. $\left[\hat{\mathbf{f}}^T_{\text{ext}}\ \hat{\boldsymbol{\tau}}^T_{\text{ext}}\right]^T$, which represents the estimated external disturbance at a given running instant, the LSTM neural network input is obtained by collecting them into the sequence $\hat{\mathbf{F}}_{\text{ext}_1}, \dots, \hat{\mathbf{F}}_{\text{ext}_n}$. Here, n is the input batch size and in the proposed case study $n = 40$. Given the corresponding classification sequence $\mathbf{S}$=(($f_{1,1}$, $f_{1,2}$, $f_{1,3}$, $f_{1,4}$),

[1] https://www.tensorflow.org/

[2] https://keras.io/

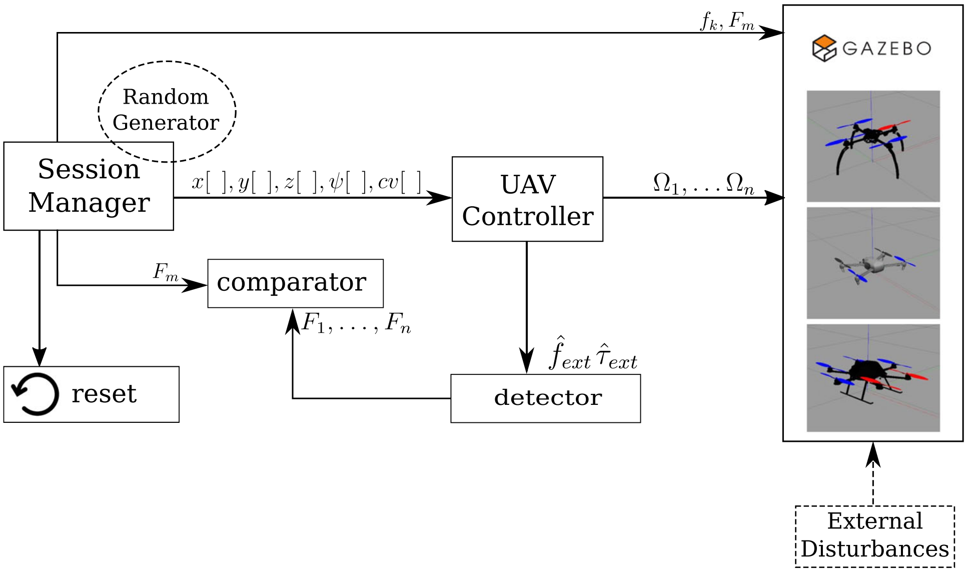


Fig. 5 Software architecture implemented to test the rotor fault detection and isolation method

Fig. 6 UAV models simulated using RotorS – (*a*): the quadcopter with the plus configuration; (*b*): the quadcopter with the cross configuration; (*c*): the hexacopter

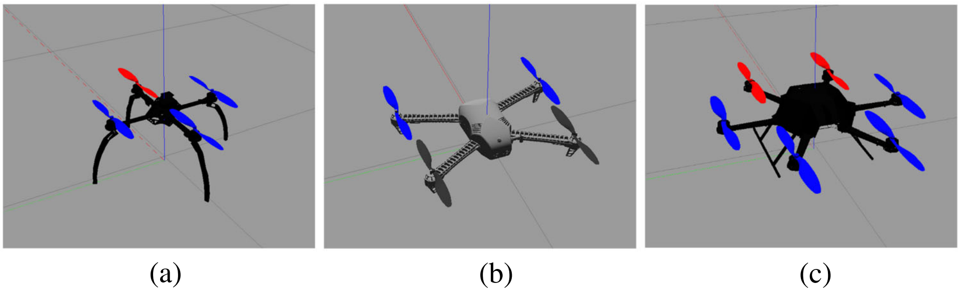


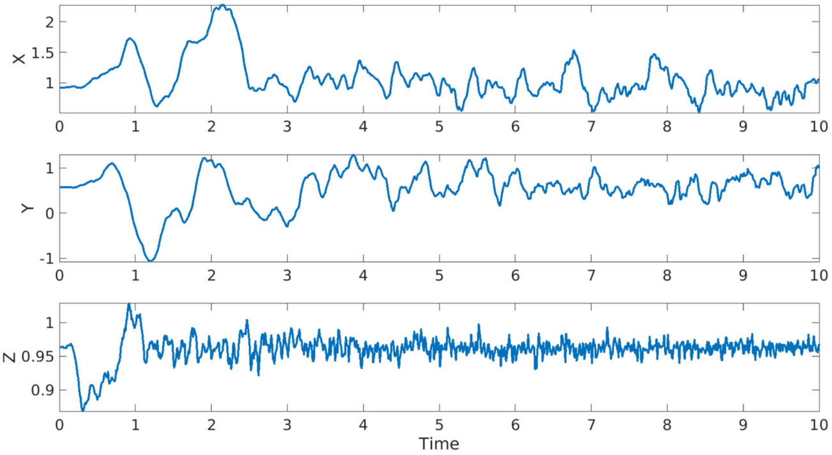
Table 6 Dynamic parameters of the simulated UAV models

Platform	m	I_x	I_y	I_z	g
Quad (+)	1.8	0.007	0.007	0.012	9.81
Quad (×)	1.51	0.034	0.0458	0.097	9.81
Hexa	1.56	0.034	0.0458	0.097	9.81

Table 7 Controller gains

Platform	k_p	k_v	k_R	k_ω
Quad (+)	6.0	5.2	2.0	0.4
Quad (×)	15	5.7	3.0	0.4
Hexa	6	4.7	3.0	0.52

Fig. 7 Wind gusts using *Dryden* turbulence model at 2.0 m altitude, 1.8 m/s wind speed and 0.3 m/s wind gust



$\ldots, (f_{n,1}, f_{n,2}, f_{n,3}, f_{n,4})$), where each 4-tuple represents the outputs related to the 4 motors status. The class c assigned to f_{tc} is the first one for which there exists a sub-sequence $(f_{t_0,c}, \ldots, f_{t_0+\Delta,c})$ such that for all $t \in [t_0, t_0 + \Delta]$, we have that $f_{t,c} > \lambda$ holds. That is, the sequence $\mathbf{f}_{ti}$ is assigned to the class c, such that the classification result c remains coherent for a fixed time windows Δ, with confidence always greater than a fixed threshold λ in that window. The use Δ parameter has been included to reduce the number of false positive data leverage on the assumption that it is better to have a delayed true positive instead of a false positive. In our case studies, Δ was empirically set to 10 steps, while λ was set to 0.6, which assures a certain amount of confidence in one class with respect to the others. A total number of 550 testing sessions are performed. Each trajectory consists of a maximum number of 4 waypoints, and the path is reset when a new fault is injected.

Despite the quality of the results obtained in the system evaluation, performance can still be increased, especially when external disturbances cause incorrect detection results. To further enhance the performance of the proposed approach, additional input data, such as the tracking error of the drone following the autonomous trajectory, could be incorporated. However, in this work, we employed the minimal amount of information that is universally available in all UAVs, whether they are operated automatically (with planned trajectories) or remotely piloted

Actuator Faults with External Disturbances

In this scenario, the multi-copter is affected by external disturbances during the detection process. This test aims to demonstrate that the fault detection method based on unmodeled dynamics data can also work in case additional external disturbances are applied to the UAV frame.

External disturbances can be caused by several phenomena: the presence of a slinging payload attached to the body of the drone, an external acting force generated by contact with the environment, or other reasons. In the following case study, these disturbances are represented as wind gusts using the *Dryden* wind model. This approach allows to comprehensively evaluate the fault detection and isolation system's robustness in varied and realistic operational conditions.

A common approach to represent atmospheric wind gusts in simulated applications uses stochastic formulations [48]. In this context, the *Dryden* wind model is a popular turbulence model to simulate wind gusts. Other works have already proposed this model to test fault detection methods [49, 50]. An example of the generated wind velocities is represented in Fig. 7. The wind velocities must be converted into wind load to apply the force on the UAV frame properly. We simplify this conversion by choosing a gain value k_w to consider the air density and the drag coefficient of the airframe. Therefore, the applied force is calculated through $\vec{F}_d = k_w \vec{v}^{\,2}$, where $\vec{v}$ is the wind speed. In the end, the desired force is applied to the centre of mass of the robotic frame using a Gazebo plugin.

Table 8 Data collected during the two testing scenarios

Platform Scenario 1	*Length* (*m*)	*Time* (min)
Quad (+)	944	116.7
Quad (×)	1317	118
Hexa	1446	127.5
Scenario 2		
Quad (+)	1430	97
Quad (×)	1315	102
Hexa	1246	121

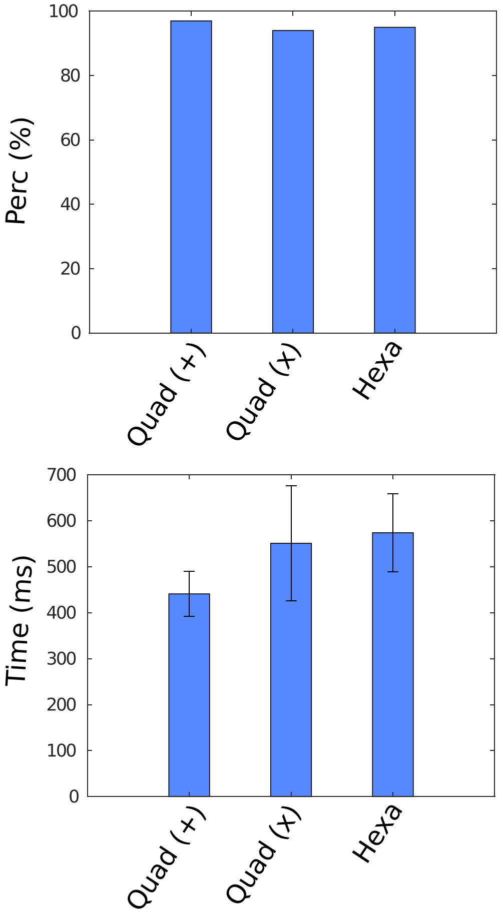


Fig. 8 Percentage of correct fault detection (top) and the mean reaction time along with its standard deviation (bottom) in scenario 1 (no external disturbances)

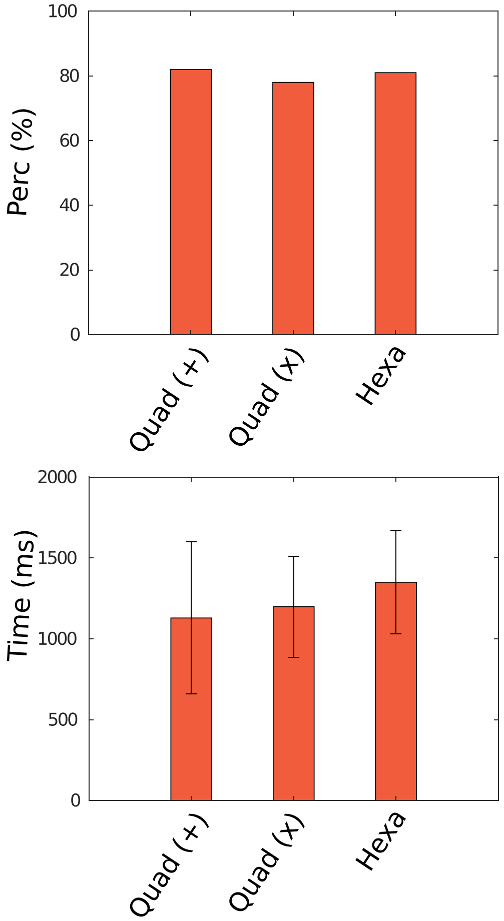


Fig. 9 Percentage of correct fault detection (top) and the mean reaction time along with its standard deviation (bottom) in scenario 2 (with external disturbances)

Results and Discussion

In Table 8, we report the total distance covered by the robots and the operative time during each scenario. The mean of the percentage of correct faults detection and the reaction time, along with its standard deviation for the two testing scenarios, are reported in Figs. 8 and 9. Results from the first scenario demonstrate that the detection system reaches a high level of accuracy ($\geq$ 90%), and it is fast enough to react to this kind of unexpected event (i.e., invoking an emergency landing operation). Similarly, the rest of the table reports results obtained from Scenario 2. In this second test case, the overall performance of the fault detection method is worse with respect to test case one. In particular, the accuracy of the classifier is weakened due to a high number of false-positive and the time needed to recognize a fault is increased. The performance degradation is motivated by wind turbulences affecting the UAV are sometimes detected as rotor faults. At the same time, the classifier cannot fix its detection results on a unique value for more than Δ iterations. At the same time, the correct value of Δ improves the detector's robustness, reducing the number of false positives.

Table 9 Comparison between the accuracy of the Data-Driven approach and the Model-Based one without external disturbances

Platform	LSTM	TSKF
Accuracy %		
Quad (+)	97.27	87.09
Quad ($\times$)	92.86	76
Mean Reaction Time (s)		
Quad (+)	0.45	3.094
Quad ($\times$)	0.55	5.794

Table 10 Comparison between the accuracy of the Data-Driven approach and the Model-Based one with external disturbances

Platform *Accuracy %*	LSTM	TSKF
Quad (+)	82.3	80.72
Quad ($\times$)	78.7	54
Mean Reaction Time (s)		
Quad (+)	1.12	3.57
Quad ($\times$)	1.19	7.74

Comparison

In order to demonstrate better performances of the approach presented in this work that uses an LSTM with respect to a model-based method, this section shows a comparison study with a two-stage Kalman filter (TSKF). The technique implemented to compare the results is the one presented in [26]. This last consists of a cascade of two Kalman filters: the former filter estimates the states of the UAV; the latter estimates and detects the fault of an actuator. Coupling equations pair these two filters. This algorithm takes as inputs the rotor velocities, linear positions and angular velocities, and it will retry a fault vector γ, which dimension is the same as the vector of the rotor velocities. This system has been implemented for the two quadrotors configurations used for this work's case studies, namely, the *Hummingbird* and *Iris*. The experiments to test this detector have been conducted in the same conditions as the previous experiments. Both methods are compared according to their *accuracy*, i.e., the percentage of correct detections over the number of simulations, and *mean reaction time*. The results of the experiments compared with our approach are shown in Table 9 for the first scenario without external disturbances. The method using LSTM overcomes the one which implements TSKF in both accuracy and mean reaction time. In particular, the best improvement can be noticed in the reaction time, which is about six times smaller. Table 10 reports the comparison results in the scenario with external disturbance. The LSTM method presented in our paper also overcomes the model-based approach in this scenario, even if the performance is worse for both according to the experiments without wind disturbance experiments.

6 Conclusion

This paper uses a novel rotor FDI method based on unmodeled disturbances for multi-copter UAVs. Our approach considers the external force acting on the UAV frame estimated through a momentum-based wrench estimator and allows us to detect partial or complete motor power loss promptly. The detection process deployed a deep neural network. In particular, different fully connected recurrent neural networks composed of LSTM cells have been designed to carry out the detection process on different UAV models. The system's training and testing have been performed using Gazebo dynamic simulator. The validity of the proposed method has been compared with a classical approach that uses a TSKF, which has been outperformed.

Future works regard the validation of this approach on a real multi-rotor platform, extending this approach to also unconventional multicopter configurations, like the actively tilting propellers UAVs.

Acknowledgements The research leading to these results has been supported by the COWBOT project, in the frame of the PRIN 2020 research program, grant n. 2020NH7EAZ_002, the AI-DROW project, in the frame of the PRIN 2022 research program, grant n. 2022BYSBYX, funded by the European Union Next-Generation EU, and the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie (grant agreement n. 953454). The authors are solely responsible for its content.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Dr. Jonathan Cacace and MSc Vincenzo Scognamiglio. The first draft of the manuscript was written by Dr. Jonathan Cacace and MSc Vincenzo Scognamiglio and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by Università degli Studi di Napoli Federico II within the CRUI-CARE Agreement. See acknowledgment section.

Code or Data Availability No code will be released for this work.

Declarations

Ethics Approval This study was performed in line with the principles of the Declaration of Helsinki. Approval was granted by the Ethics Committee of University of Naples "Federico II".

Consent to Partecipate Not applicable.

Consent for Publication Not applicable.

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

References

1. Amazon.com: Prime Air. https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011 (2024). Accessed 29 March 2021
2. Zhao, S., Ruggiero, F., Fontanelli, G.A., Lippiello, V., Zhu, Z., Siciliano, B.: Nonlinear model predictive control for the stabilization of a wheeled unmanned aerial vehicle on a pipe. IEEE Robot. Autom. Lett. **4**(4), 4314–4321 (2019). https://doi.org/10.1109/LRA.2019.2931821
3. Trujillo, M.A., Martinez-de Dios, J.R., Martín, C., Viguria, A., Ollero, A.: Novel aerial manipulator for accurate and robust industrial ndt contact inspection: A new tool for the oil and gas inspection industry. Sensors 19(6) (2019)
4. Venkataraman, R., Bauer, P., Seiler, P., Vanek, B.: Comparison of fault detection and isolation methods for a small unmanned aircraft. Control Eng. Pract. **84**, 365–376 (2019)
5. Ruggiero, F., Lippiello, V., Ollero, A.: Aerial manipulation: A literature review. IEEE Robot. Autom. Lett. **3**(3), 1957–1964 (2018)
6. Funahashi, K.-I., Nakamura, Y.: Approximation of dynamical systems by continuous time recurrent neural networks. Neural Netw. **6**(6), 801–806 (1993)
7. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: Lstm: A search space odyssey. IEEE Trans. Neural Netw. Learn. Syst. **28**(10), 2222–2232 (2017)
8. Furrer, F., Burri, M., Achtelik, M., Siegwart, R.: RotorS - A Modular Gazebo MAV Simulator. Framework **625**, 595–625 (2016)
9. Guo, K., Liu, L., Shi, S., Liu, D., Peng, X.: Uav sensor fault detection using a classifier without negative samples: A local density regulated optimization algorithm. Sensors 19(4) (2019)
10. Sun, R., Cheng, Q., Wang, G., Ochieng, W.Y.: A novel online data-driven algorithm for detecting uav navigation sensor faults. Sensors 17(10) (2017)
11. Aboutalebi, P., Abbaspour, A., Forouzannezhad, P., Sargolzaei, A.: A novel sensor fault detection in an unmanned quadrotor based on adaptive neural observer. J. Intell. Robot. Syst. 90 (2018)
12. Gilmore, J.P., McKern, R.A.: A redundant strapdown inertial reference unit (siru). J. Spacecr. Rocket **9**(1), 39–47 (1972)
13. Saied, M., Lussier, B., Fantoni, I., Shraim, H., Francis, C.: Active versus passive fault-tolerant control of a redundant multirotor uav. Aeronaut. J. **124**(1273), 385–408 (2020)
14. Baskaya, E., Hamandi, M., Bronz, M., Franchi, A.: A novel robust hexarotor capable of static hovering in presence of propeller failure. IEEE Robot. Autom. Lett. **6**(2), 4001–4008 (2021)

15. Mazeh, H., Saied, M., Shraim, H., Francis, C.: Fault-tolerant control of an hexarotor unmanned aerial vehicle applying outdoor tests and experiments. IFAC-PapersOnLine **51**(22), 312–317 (2018). 12th IFAC Symposium on Robot Control SYROCO 2018
16. Mueller, M.W., D'Andrea, R.: Relaxed hover solutions for multicopters: Application to algorithmic redundancy and novel vehicles. Int. J. Robot. Res. **35**(8), 873–889 (2016)
17. Stephan, J., Schmitt, L., Fichter, W.: Linear parameter-varying control for quadrotors in case of complete actuator loss. J. Guid. Control Dyn. **41**(10), 2232–2246 (2018)
18. Lippiello, V., Ruggiero, F., Serra, D.: Emergency landing for a quadrotor in case of a propeller failure: A pid based approach. In: 2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014), pp. 1–7 (2014)
19. Pourpanah, F., Zhang, B., Ma, R., Hao, Q.: Anomaly detection and condition monitoring of uav motors and propellers. 2018 IEEE SENSORS, 1–4 (2018)
20. Iannace, G., Ciaburro, G., Trematerra, A.: Fault diagnosis for uav blades using artificial neural network. Robotics 8(3) (2019)
21. Sharifi, F., Mirzaei, M., Gordon, B.W., Zhang, Y.: Fault tolerant control of a quadrotor uav using sliding mode control. In: 2010 Conference on Control and Fault-Tolerant Systems (SysTol), pp. 239–244 (2010)
22. Freddi, A., Longhi, S., Monteriù, A.: A diagnostic thau observer for a class of unmanned vehicles. J. Intell. Robot. Syst. **67**, 61–73 (2012)
23. Cen, Z., Noura, H.: An adaptive thau observer for estimating the time-varying loe fault of quadrotor actuators. In: 2013 Conference on Control and Fault-Tolerant Systems (SysTol), pp. 468–473 (2013)
24. Avram, R.C., Zhang, X., Muse, J.: Quadrotor actuator fault diagnosis and accommodation using nonlinear adaptive estimators. IEEE Trans. Control Syst. Technol. **25**(6), 2219–2226 (2017). https://doi.org/10.1109/TCST.2016.2640941
25. Caliskan, F., Hacizade, C.: Sensor and actuator fdi applied to an uav dynamic model. IFAC Proc. Volumes **47**(3), 12220–12225 (2014). https://doi.org/10.3182/20140824-6-ZA-1003.01013 . 19th IFAC World Congress
26. Amoozgar, M.H., Chamseddine, A., Zhang, Y.: Experimental test of a two-stage kalman filter for actuator fault detection and diagnosis of an unmanned quadrotor helicopter. J. Intell. Robotics Syst. **70**(1–4), 107–117 (2013). https://doi.org/10.1007/s10846-012-9757-7
27. Tzoumanikas, D., Yan, Q., Leutenegger, S.: Nonlinear mpc with motor failure identification and recovery for safe and aggressive multicopter flight. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 8538–8544 (2020). https://doi.org/10.1109/ICRA40945.2020.9196690
28. Guo, D., Wang, Y., Zhong, M., Zhao, Y.: Fault detection and isolation for unmanned aerial vehicle sensors by using extended pmi filter. IFAC-PapersOnLine **51**(24), 818–823 (2018). 10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2018
29. Fourlas, G.K., Karras, G.C.: A survey on fault diagnosis methods for uavs. In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 394–403 (2021). https://doi.org/10.1109/ICUAS51884.2021.9476733
30. De Luca, A., Albu-Schaffer, A., Haddadin, S., Hirzinger, G.: Collision detection and safe reaction with the dlr-iii lightweight manipulator arm. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1623–1630 (2006)
31. Ruggiero, F., Cacace, J., Sadeghian, H., Lippiello, V.: Impedance control of vtol uavs with a momentum-based external generalized forces estimator. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 2093–2099 (2014)
32. Ruggiero, F., Cacace, J., Sadeghian, H., Lippiello, V.: Passivity-based control of vtol uavs with a momentum-based estimator of external wrench and unmodeled dynamics. Robot. Auton. Syst. **72**, 139–151 (2015)
33. Guo, D., Zhong, M., Ji, H., Liu, Y., Yang, R.: A hybrid feature model and deep learning based fault diagnosis for unmanned aerial vehicle sensors. Neurocomputing **319**, 155–163 (2018)
34. Samy, I., Postlethwaite, I., Gu, D.-W.: Sensor fault detection and accommodation using neural networks with application to a nonlinear unmanned air vehicle model. Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng. **224**(4), 437–447 (2010)
35. Wen, L., Li, X., Gao, L., Zhang, Y.: A new convolutional neural network-based data-driven fault diagnosis method. IEEE Trans. Ind. Electron. **65**(7), 5990–5998 (2018)
36. Zhang, X., Zhao, Z., Wang, Z., Wang, X.: Fault detection and identification method for quadcopter based on airframe vibration signals. Sensors **21**(2) (2021)
37. Sadhu, V., Zonouz, S., Pompili, D.: On-board deep-learning-based unmanned aerial vehicle fault cause detection and identification. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 5255–5261 (2020)
38. Wang, B., Liu, D., Peng, Y., Peng, X.: Multivariate regression-based fault detection and recovery of uav flight data. IEEE Trans. Instrum. Meas. **69**(6), 3527–3537 (2020)
39. Madani, T., Benallegue, A.: Backstepping control for a quadrotor helicopter. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3255–3260 (2006)
40. Lee, T., Leok, M., McClamroch, N.H.: Geometric tracking control of a quadrotor uav on se(3). In: 49th IEEE Conference on Decision and Control (CDC), pp. 5420–5425 (2010)
41. Ruggiero, F., Trujillo, M.A., Cano, R., Ascorbe, H., Viguria, A., Peréz, C., Lippiello, V., Ollero, A., Siciliano, B.: A multilayer control for multirotor uavs equipped with a servo robot arm. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 4014–4020 (2015)
42. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–80 (1997)
43. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. Proceedings of the 30th International Conference on International Conference on Machine Learning - vol. 28, pp. 1310–1318 (2013)
44. jMavSim. https://github.com/PX4/jMAVSim. (2024) Accessed 29 March 2021
45. Shah, S., Dey, D., Lovett, C., Kapoor, A.: AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles (2017)
46. Ferrera, E., Alcántara, A., Capitán, J., Castaño, A.R., Marrón, P.J., Ollero, A.: Decentralized 3d collision avoidance for multiple uavs in outdoor environments. Sensors **18**(12) (2018)
47. Suárez Fernández, R.A., Dominguez, S., Campoy, P.: L1 adaptive control for wind gust rejection in quad-rotor uav wind turbine inspection. In: 2017 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1840–1849 (2017)
48. Vann, F.W.: Gust loads on aircraft: Concepts and applications. f. m. hoblit. american institute of aeronautics and astronautics, washington, d.c. 1989. 306 pp. illustrated. 39.95$(aiaamembers)$49.95 (non-members). Aeronaut. J. (1968) **93**(930), 406–406 (1989)
49. Abichandani, P., Lobo, D., Ford, G., Bucci, D., Kam, M.: Wind measurement and simulation techniques in multi-rotor small unmanned aerial vehicles. IEEE Access **8**, 54910–54927 (2020)
50. Zhong, Y., Zhang, Y., Zhang, W., Zuo, J., Zhan, H.: Robust actuator fault detection and diagnosis for a quadrotor uav with external disturbances. IEEE Access **6**, 48169–48180 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Jonathan Cacace was born in Naples, Italy, on December 13, 1987. He earned a master's degree in computer science, graduating magna cum laude in 2012, and later completed a Ph.D. in robotics in 2016, both from the University of Naples Federico II. From 2019 to 2022, he worked as an assistant professor at the same university, where he taught advanced robotics programming and mobile robotics courses. His research focuses on human-robot interaction in Industry 4.0, autonomous UAV control for inspection and maintenance, and robotic manipulation. Jonathan is actively involved in the IEEE community, serving as an Associate Editor for various conferences and journals, and contributes to editorial boards for major robotics conferences. Currently, he is a senior researcher in cognitive and social robotics at Eurecat - Technology Centre of Catalonia.

Vincenzo Scognamiglio was born in Sarno, Italy, on November 1996. He received the MSc. degree in Automation Engineering in 2021 and since that year is Ph.D. candidate at the Department of Electrical Engineering and Information Technology at University of Naples Federico II. His research interests rely on autonomous navigation and state estimation of aerial robots.

Fabio Ruggiero is Associate Professor of Automatic Control and Robotics in the Department of Electrical Engineering and Information Technology at University of Naples Federico II, where he is responsible for the DynLeg (Dynamic manipulation and Legged robotics) research area. His research interests are focused on control strategies for dexterous, dual-hand and nonprehensile robotic manipulation, aerial robots, aerial manipulators, and legged robots. He is associate editor of the IEEE Transaction on Robotics. He is Chair of the IEEE Italy RAS Chapter. He has published more than 110 journal articles, conference papers, and book chapters. He has participated to several European research projects, with the role of WP leader. He has been principal investigator of four projects funded by the Italian Ministry of Research.

Vincenzo Lippiello (SM'17) was born in Naples, Italy, in 1975. He received the Laurea degree in electronic engineering and the Research Ph.D. degree in information engineering from the University of Naples Federico II, Naples, in 2000 and 2004, respectively, where he is currently an Associate Professor of automatic control with the Department of Electrical Engineering and Information Technology. He has authored or coauthored more than 120 journal and conference papers and book chapters. His research interests include visual serving of robot manipulators, hybrid visual/force control, adaptive control, grasping and manipulation, aerial robotics, and visual object tracking and reconstruction.