Fault Detection and Isolation for a Standard Quadrotor Using a Deep Neural Network Trained on a Momentum-based Estimator

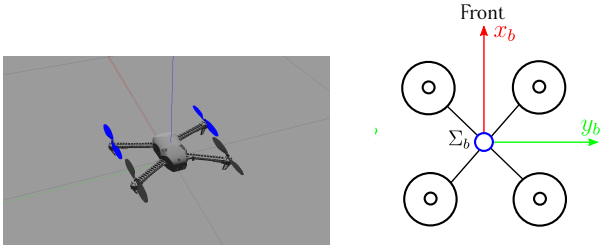
Vincenzo Scognamiglio[1], Jonathan Cacace[2], Fabio Ruggiero[1], Vincenzo Lippiello[1]

***Abstract*— Rapid identification of motor failures holds significant importance for ensuring the safety of multi-rotor unmanned aerial vehicles. This study introduces a method for detecting and isolating motor faults in standard quadrotors, utilizing an external wrench estimator and a recurrent neural network with long short-term memory nodes. The proposed approach treats partial or complete motor failure as an external disturbance affecting the quadrotor. Consequently, the external wrench estimator trains the network to quickly discern whether the estimated wrench results from a motor fault, identifying the specific motor involved, or if it stems from unmodeled dynamics or external factors such as wind or contacts. The training and testing of this method were conducted in a simulation environment equipped with a physics engine.**

I. INTRODUCTION

In recent years, there has been a notable rise in the deployment of aerial vehicles for a variety of service-oriented tasks. Among these, Vertical Take-Off and Landing (VTOL) unmanned aerial vehicles (UAVs) have emerged as particularly well-suited for assignments such as inspection, surveillance, and search and rescue missions. Their agility, swift maneuverability, and hovering capabilities make them ideal for such operations. In today's landscape, e-commerce companies are exploring UAVs for doorstep delivery of goods, while safety-critical environments demand their presence for tasks like inspecting oil and gas facilities, where drones can perform visual and contact inspections of fluid-carrying pipelines [1], [2] or installation of bird diverters on transmission lines [3], [4]. Safety considerations are paramount in these domains to mitigate potential harm to human operators and prevent damage to critical equipment. The actuation system of a multi-copter consists of various brushless motors with either fixed or actuated propellers. Any damage to the UAV's propellers can significantly reduce its propulsion system, compromising its stability during flight. Therefore, implementing a fault detection system is crucial for ensuring the safety of multi-rotor operations [5], serving as the primary motivation for this study.

Detecting motor faults in a quadrotor UAV poses a unique challenge, primarily because quadrotor rotors typically lack feedback mechanisms. Consequently, conventional failure detection techniques rely on measurements from onboard sensors.



(a) (b)

Fig. 1: Quadrotor model used in this work. *(a)* - UAV Iris simulated using RotorS; *(b)* - the quadrotor cross frame configuration.

This study utilizes a deep neural network, which is supplied with estimated disturbances, to identify and isolate potential motor faults specific to standard flat quadrotor UAVs. Throughout the quadrotor's flight, various unmodeled aerodynamic disturbances impact its behavior and stability. These disturbances are commonly perceived as the result of an aggregated external wrench applied at the quadrotor's center of mass. Specialized estimators are devised to enhance the tracking performance of onboard controllers.

The core idea behind this research is that a malfunctioning motor in the quadrotor, whether partially or completely, introduces an additional disturbance that affects the tracking controller's performance. Existing estimators are unable to discern between the distinct external effects contributing to the consolidated external wrench at the quadrotor's center of mass. Hence, it is imperative to develop an auxiliary system that collaborates with the estimator to differentiate whether the estimated disturbance stems from a partial or complete failure of a quadrotor motor. In this context, the supplementary system takes the form of a recurrent neural network (RNN) [6] consisting of long short-term memory (LSTM) [7] nodes, which analyze the estimated values.

Data-driven and machine-learning approaches have been independently utilized to tackle actuator Fault Detection and Isolation (FDI) issues in UAVs [8], [9], [10]. Similarly, in our approach, a deep recurrent neural network is proposed to identify potential faults. The broader application of deep learning techniques is evident in various scenarios. For instance, in [11], authors deployed an LSTM network to predict UAV malfunctions by analyzing platform vibrations. In [12], a deep learning approach was used to identify causes of failure following crashes or incidents. Additionally,

*This work was not supported by any organization

[1]Vincenzo Scognamiglio, Fabio Ruggiero and Vincenzo Lippiello are with PRISMA Lab and CREATE Consortium, Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy. vincenzo.scognamiglio2@unina.it

[2]Jonathan Cacace is with with Eurecat, Centre Tecnològic de Catalunya, Robotics and Automation Unit, jonathan.cacace@eurecat.org

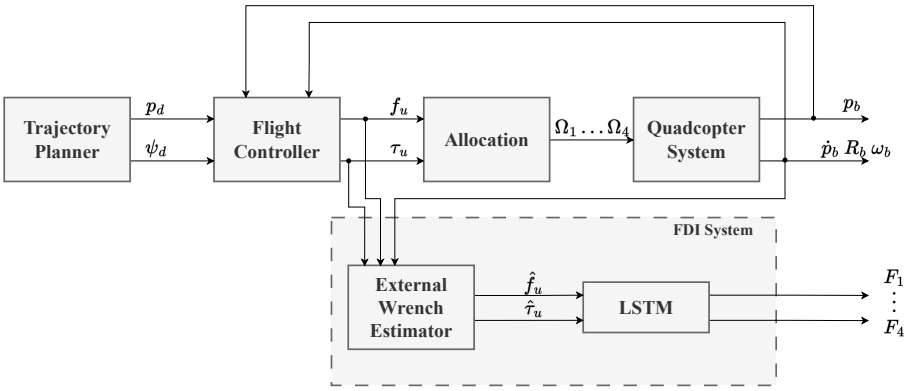


Fig. 2: System architecture. In the dashed box are included the two modules comprising the quadrotor fault detection and identification method proposed in this work. The external wrench estimator computes unmodeled forces and torques acting on the quadrotor body leveraging on commanded forces and torques, that came from the flight controller, linear and angular velocity, and orientation in the body frame.

in [13], a data-driven approach based on LSTM networks was employed to detect onboard sensor drift. In contrast, our approach focuses on estimating the presence of external forces, such as a loss of thrust from one of the motors, to accurately identify a fault. The network is specifically trained to identify and isolate faults during UAV flights involving unpredictable trajectories and varied external disturbances, such as randomly induced wind effects and motor faults. To validate the efficacy of the proposed FDI approach, extensive testing has been conducted within the Gazebo simulator on a standard quadrotor configuration (see Fig. 1). The simulation leverages the RotorS ROS [14] package to replicate the dynamics of the quadrotor and the behavior of its motors. This ROS package facilitates the simulation of quadrotor dynamics and the interactions of its propellers, ensuring a comprehensive evaluation of the developed FDI approach.

In summary, the contributions provided include:

- recognizing the rotor fault as an external disturbance on the quadrotor and estimating the lumped wrench on its center of mass;
- implementing a data-driven system capable of determining whether the estimated wrench originates from a (partial or complete) rotor fault and identifying the fault source;
- conducting a simulation campaign employing a quadrotor model.

The remainder of the paper is organized as follows: Sec. II a preliminary description of the quadrotor model is presented, along with the momentum-based external wrench estimator; in Sec. III the innovative approach that exploits recurrent neural network is presented, a description of the training session of the network is reported too; Sec. IV the extensive simulation campaign is illustrated for both scenarios (with and without external disturbances) and in Sec. V the results are presented and commented; finally, Sec. VI presents conclusions and possible future developments.

II. SYSTEM DESCRIPTION

The proposed methodology involves estimating unmodeled disturbances affecting the quadrotor frame through a momentum-based external wrench estimator [15]. This estimator, previously applied to quadrotors to enhance position control loop performance by compensating for detected disturbance effects [16], [17], [18], is employed in this study solely for characterizing rotor power loss. A data-driven approach is incorporated to discern whether the estimated disturbance stems from a partial or complete failure of a quadrotor motor, the complete system architecture scheme is in Fig. 2. In this section, the dynamic model of a multi-rotor is presented along with the wrench estimator used.

A. Dynamic Models of Multi-Rotors

Flat multi-rotors are characterized as under-actuated systems, with six degrees of freedom in the mathematical model under consideration, yet only possessing four control inputs. Consider two frames, Σ_w and Σ_b, representing the world-fixed and body-fixed frames attached to the UAV center of mass, respectively. The position of Σ_b in the world-fixed frame is denoted as $\mathbf{p}_b = [x\ y\ z]^T \in \mathbb{R}^3$, while its attitude is represented by the rotation matrix $\mathbb{R}_b \in SO(3)$. In this context, the dynamic equations of the UAV can be expressed as follows

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

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

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

where $\mathbf{e}_3 = [0\,0\,1]^T \in \mathbb{R}^3$, $m > 0$ represents the multi-rotor's mass, $\mathbf{I}_b \in \mathbb{R}^{3\times3}$ is its symmetric and positively definite inertia matrix expressed in Σ_b, $\boldsymbol{\omega}_b^b \in \mathbb{R}^3$ denotes the angular velocity vector expressed in Σ_b, $S(\cdot) \in \mathbb{R}^{3\times3}$ as the skew-symmetric operator, $u_T \in \mathbb{R}$ and $\boldsymbol{\tau}^b = [\,\tau_x\ \tau_y\ \tau_z\,]^T \in \mathbb{R}^3$ represent the total thrust force and control torques, respectively. Additionally, $\mathbf{f}_{ext} \in \mathbb{R}^3$ and $\boldsymbol{\tau}_{ext}^b \in \mathbb{R}^3$ are lumped vectors indicating unknown forces and torques (expressed in Σ_b),

such as aerodynamic effects, flapping dynamics, parametric uncertainties, imbalances arising from batteries and/or on-board sensors, wind gusts, interactions with the environment, and propeller faults, among other factors. The translation of the desired force and torque into rotational velocities for the multi-rotor propellers, essential for maneuvering the aerial platform, is contingent upon the chosen frame. More specifically, it relies on factors such as the number of motors and their respective positions on the frame.

Considering a common quadrotor with a cross configuration (see Fig. 1b), where $\Omega_i \in \mathbb{R}$ for $i = 1, ..., 4$ represents the rotational velocities of the individual propellers, the control input vector for the quadrotor, denoted as $\mathbf{u} = [\, u_T \;\; \tau^{b^T} \,]^T$, can be computed using the following expression [19]

$$\mathbf{u} = \begin{bmatrix} c_t & c_t & c_t & c_t \\ \sin(\alpha_1)\, l_1\, c_t & \sin(\alpha_2)\, l_2\, c_t & \sin(\alpha_3)\, l_3\, c_t & \sin(\alpha_4)\, l_4\, c_t \\ \cos(\alpha_1)\, l_1\, c_t & \cos(\alpha_2)\, l_2\, c_t & \cos(\alpha_3)\, l_3\, c_t & \cos(\alpha_4)\, l_4\, c_t \\ d\, c_a & d\, c_a & d\, c_a & d\, c_a \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}, \tag{2}$$

where $\alpha_i \in \mathbb{R}$ denotes the angle of the rotor on the quadrotor frame concerning Σ_b, $l_i > 0$ represents the distance between the i-th propeller and the origin of Σ_b, $c_t > 0$ and $c_a > 0$ are the thrust and torque coefficients of the propellers, respectively, and d indicates the rotation direction of the motor, being positive for clockwise and negative for counterclockwise rotor rotations.

B. External Wrench Estimator

The objective of the *External Wrench Estimator* module is to assess and estimate the unmodeled disturbances influencing the quadrotor frame. It incorporates a momentum-based estimator previously applied in [16], [20], [18] for enhancing the position control of a quadrotor. The module takes as input the desired control force and torque (f_u, τ_u), along with 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 $\hat{\mathbf{F}}_{ext}(t) = \begin{bmatrix} \hat{\mathbf{f}}_{ext}^T(t) & \hat{\tau}_{ext}^{b^T}(t) \end{bmatrix}^T$ are determined through the following calculations using the second-order proposed estimator

$$\hat{\mathbf{F}}_{ext}(t) = \mathbf{K}_1 \left(\int_0^t -\hat{\mathbf{F}}_{ext}(\sigma) + \mathbf{K}_2 \left(\alpha(\sigma) - \int_0^t \left(\begin{bmatrix} u_T \mathbf{R}_b \mathbf{e}_3 - mg\mathbf{e}_3 \\ \tau^b - \mathbf{S}(\omega_b^b)\mathbf{I}_b \omega_b^b \end{bmatrix} + \hat{\mathbf{F}}_{ext}(\sigma) \right) \mathrm{d}\sigma \right) \mathrm{d}\sigma \right). \tag{3}$$

Here, the matrices $\mathbf{K}_1 \in \mathbb{R}^{6\times6}$ and $\mathbf{K}_2 \in \mathbb{R}^{6\times6}$ are positive-definite, and $\alpha \in \mathbb{R}^6$ represents the system's momentum as

$$\alpha = \begin{bmatrix} m\mathbf{I}_3 & \mathbf{O}_3 \\ \mathbf{O}_3 & \mathbf{I}_b \end{bmatrix} \begin{bmatrix} \dot{\mathbf{p}}_b \\ \omega_b^b \end{bmatrix}. \tag{4}$$

This study introduces a pivotal innovation by training a neural network to differentiate cases where the module's output corresponds to a fault in the propellers, thereby identifying the specific propeller responsible for the issue. This approach contrasts with merely attributing the issue to an external lumped wrench originating from other sources.

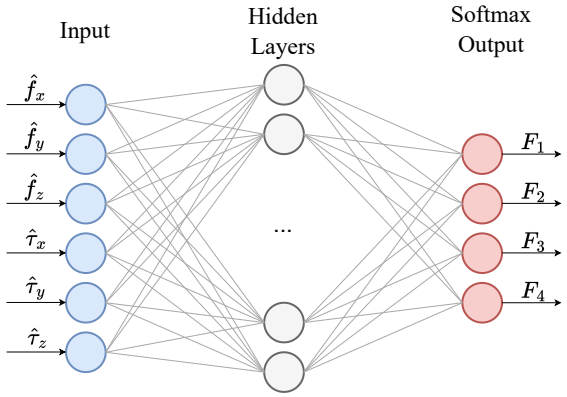


Fig. 3: LSTM Neural network structure.

III. ROTOR FAULT DETECTION AND ISOLATION

In this section, we elaborate on the motor fault detection and isolation process. Our proposed approach involves an RNN based on LSTM [21], specifically designed to address the vanishing gradient problem encountered in traditional RNNs [22]. Incorporating an LSTM structure replaces the conventional hidden unit with a memory cell, addressing challenges associated with information propagation. This enhancement is facilitated by three gates: the input gate, forget gate, and output gate, which control the storage and retrieval of memory, discerning between pertinent and extraneous information. Leveraging LSTM networks, renowned for their proficiency in analyzing time-series data, aligns well with the requirements of our application.

The input layer comprises six nodes, representing the three-dimensional forces and torques estimated by the *External Wrench Estimator* module (i.e., $\hat{\mathbf{f}}_{ext}$, $\hat{\tau}_{ext}$). The output layer's size is contingent on the number of motors in the quadrotor. Each output node assesses the operating status of a motor on the aerial platform. The neural network structure, illustrated in Figure 3, includes an input layer, a hidden layer of LSTM cells, and an output layer associated with a *softmax* function. Therefore, a network with 4 output nodes for detecting rotor faults in the quadrotor model is deployed. A hidden layer comprising 25 nodes is also considered. The network training process involved utilizing 100 epochs for the data batch, with the batch size experimentally set to 20 samples.The *Adam* optimizer and the *categorical cross-entropy* loss function were employed for effective model training.

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

TABLE I: Set of parameters used to generate the training dataset.

Network training

In the proposed application, the estimation of unmodeled disturbances in the presence of a motor fault is intricately tied to the system dynamics and rotor configuration. Since the UAV configuration taken into account in this work is the standard cross quadrotor, a dataset has been generated, simulating various faulty trajectories and encompassing a range of static and random parameters. Table I provides a comprehensive list of these parameters along with their boundary values. For the considered platform, a new trajectory is charted, incorporating random target way-points ($[x,y,z]$ and ψ) and cruise velocity (cv). The target is generated utilizing random functions from standard C++ libraries based on a uniform distribution. Simultaneously, during flight, a decision is randomly made to inject a fault into one of the motors of the aerial platform. This decision is governed by comparing a generated number (g_{th}) to a predetermined threshold (e.g., 500 in the proposed training sessions). When the generated number exceeds the threshold, a new fault is triggered. The selection of this threshold, determined experimentally, ensures a balanced flight time with and without motor faults. Here, ω_i represents the velocity of a quadrotor propeller, and $f_k\,\omega_i$ simulates the loss of power in motor m. Similar to target way-points, both f_k and m are randomly chosen during navigation. In each training session, the recorded data includes both disturbances and a motor status vector, represented by $\mathbf{f} = [f_1,\ldots,f_n]$, comprising n binary values. Here, n corresponds to the number of motors on the aircraft, and f_i is set to one if a fault is introduced in the i-th motor, and zero otherwise. The estimation process operates at a frequency of 100 Hz, with data being stored in the dataset at the same frame rate. It is essential to note that, for a given timeframe, only one rotor fault was considered. The conclusion of a training session is triggered either by a predefined duration following the injection of a motor fault or when the fault reaches a critical state, such as complete loss of spinning force, jeopardizing platform stabilization. The latter condition is identified by monitoring the attitude error of the onboard controller. If this error surpasses its control saturation threshold (e.g., 0.35 rad), the system is reset, initiating a new training session. Subsequently, the collected datasets are partitioned into training and test sets, with 70% and 30% coverage, respectively. Using the conventional definitions where *true/false* positive denotes correct/wrong identification of a motor fault (TP/FP), and *true/false* negative signifies correct/wrong classification of normal working conditions for UAV motors (TN/FN), it is possible to define three quantities to define the quality of the trained network. These indexes are defined as follows:

- Precision (P) measures the ratio of correct predictions to total predictions:

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

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

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

- Accuracy (Acc) gauges the correct predicted data over all elements in the dataset:

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

Higher values for P, R, and Acc indicate better performance of the FDI method. The result of the test made considering the standard flat quadrotor with cross configuration and using the test dataset are the following $P = 0.95$, $R = 0.81$ and $Acc = 0.92$. Finally, Table II presents the confusion matrix, illustrating that the deployed LSTM network effectively isolates faults and accurately identifies damaged rotors. Furthermore, these results are based on the samples stored in the datasets. In a more realistic scenario, false-negative phenomena can be mitigated by conducting over-time testing sessions, as validated in the next section.

	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

TABLE II: Confusion matrix for the standard flat quadrotor flying with the cross configuration.

IV. EXPERIMENTS

This section details the evaluation of the system through simulated case studies aimed at showcasing the effectiveness of the proposed FDI method. Various tools are available for simulating aerial robot dynamics, and for this work, we leverage RotorS, a ROS-based simulator offering a modular framework for designing Micro Aerial Vehicles to test control and state estimation algorithms. The tests were conducted on a standard computer running Ubuntu 20.04 GNU/Linux OS and ROS Noetic as the robotic middleware. The LSTM network was implemented using the TensorFlow library[1] through the Keras[2] high-level interface programmed in Python. The UAV model is inspired by the *Iris* quadrotor from *3D Robotics* (see Figure 1b). The dynamic parameters of the simulated UAV are presented in Table III. During the simulation case studies, the quadrotor follows a takeoff procedure to reach a fixed altitude. Similar to the training sessions, a series of waypoints are randomly generated to facilitate the execution of various trajectories. Throughout this motion, rotor faults may occur at any point. The initiation of a new test involves the *Session Manager*, tasked with establishing a fresh trajectory defined by parameters such as x, y, z, ψ, and cv. Concurrently, this module introduces a fault into one of the quadrotor's motors. The *detector* module loads the LSTM-trained model corresponding to the

[1] https://www.tensorflow.org/
[2] https://keras.io/

tested UAV and continuously assesses the state of the rotors. Upon detecting a new fault, the *detector* module's output is compared with that of the *Session Manager*, enabling the characterization of correct or incorrect classification results. Subsequent to the fault injection, the simulation scene is reset to initiate a new testing session.

	UAV parameters
m	1.51
I_x	0.034
I_y	0.0458
I_z	0.097
g	9.81

TABLE III: Dynamic parameters of the simulated UAV model.

Two scenarios were explored to assess the effectiveness of the detection system: one with external disturbances and one without. In the first scenario, random trajectories were executed to mirror the conditions encountered during the training phase of the *LSTM* network. In the second scenario, external forces were introduced to affect the flying platform with turbulence noise. These disturbances simulate wind gusts, a major external disturbance source for quadrotors operating outdoors. For each test, the following data were recorded:

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

A. Experiments without external disturbances

Given $\hat{\mathbf{F}}_{ext}$, denoted as $\begin{bmatrix}\hat{\mathbf{f}}_{ext}^T & \hat{\tau}_{ext}^T\end{bmatrix}^T$, representing the estimated external disturbance at a given moment, the LSTM neural network input is formed by organizing them into the sequence $\hat{\mathbf{F}}_{ext_1},\ldots,\hat{\mathbf{F}}_{ext_n}$. Here, n denotes the input batch size, and in our proposed case study, $n = 40$. Correspondingly, we have the classification sequence $\mathbf{S}$=$((f_{1,1},f_{1,2},f_{1,3},f_{1,4}),\ldots,(f_{n,1},f_{n,2},f_{n,3},f_{n,4}))$, where each 4-tuple represents the outputs related to the status of the 4 motors. The class c assigned to f_{tc} is determined by the first one for which a sub-sequence $(f_{t_0,c},\ldots,f_{t_0+\Delta,c})$ exists such that for all $t \in [t_0, t_0+\Delta]$, $f_{t,c} > \lambda$ holds. In other words, the sequence $\mathbf{f}_{ti}$ is assigned to class c, ensuring that the classification result c remains consistent for a fixed time window Δ, with confidence consistently exceeding a set threshold λ within that window. The parameter Δ is introduced to minimize the impact of false positives, operating on the assumption that delaying a true positive is preferable to a false positive. In our case studies, Δ was empirically set to 10 steps, while λ was set to 0.6, ensuring a certain level of confidence in one class compared to the others. A total of 550 testing sessions were conducted, where each trajectory comprised a maximum of 4 waypoints, and the path reset upon injecting a new fault. The results of this simulation are collected in Table IV under the *Scanario 1* label. To evaluate the performance of the proposed method the *Mean Reaction Time (MRT)* index has been introduced, representing the average time of reaction of the proposed system computed over the 550 testing sessions.

B. Experiments with external disturbances

In this scenario, the quadrotor undergoes external disturbances during the fault detection process. This test is designed to illustrate that the fault detection method, relying on unmodelled dynamics data, remains effective even when additional external disturbances are introduced to the quadrotor frame. These disturbances are simulated as wind gusts utilizing the *Dryden* wind model. A commonly employed strategy to simulate atmospheric wind gusts in simulated environments involves stochastic formulations [23]. To properly apply force to the quadrotor frame, these wind velocities need to be converted into wind load. We simplify this conversion by incorporating a gain value k_w (set to, for example, 3), accounting for air density and the drag coefficient of the airframe. Consequently, the applied force is computed as $\vec{F}_d = k_w \vec{v}^{\,2}$, where $\vec{v}$ denotes the wind speed. Finally, this desired force is applied to the center of mass of the robotic frame using a Gazebo plugin.

V. RESULTS AND DISCUSSION

In Table IV, the results of the extensive simulation campaign for both scenarios are summarized. It presents the mean percentage of accurate fault detections, reaction times, total distance covered by the drone, and the total time of the testing session. Results from the first scenario demonstrate a high level of accuracy (approximately 93%) in the detection system, reacting promptly to unexpected events and initiating emergency landing operations. Conversely, Scenario 2 exhibits a comparatively reduced overall performance in fault detection. This decline is attributed to a notable increase in false positives, as wind turbulence affecting the quadrotor are occasionally misinterpreted as rotor faults. Additionally, the classifier faces challenges in stabilizing detection results consistently for more than Δ iterations. The appropriate selection of Δ, while enhancing detector robustness, contributes to a reduction in false positives.

Results	
Scenario 1	
Length (*m*)	1317
Time (min)	118
Accuracy (%)	92.86
MRT (*s*)	0.55
Scenario 2	
Length (*m*)	1315
Time (min)	102
Accuracy (%)	78.7
MRT (*s*)	1.19

TABLE IV: Data collected during the two testing scenarios.

VI. CONCLUSIONS

In this study, we unveil a rotor FDI method custom-tailored for the standard flat quadrotor. Our approach incorporates external forces acting on the quadrotor frame, meticulously estimated through a momentum-based wrench estimator. This novel methodology ensures swift and accurate detection of both partial and complete motor power loss instances, paving the way for enhanced UAV safety and reliability. At the heart of our detection process lies a deep neural network, crafted to discern subtle variations indicative of rotor faults. Leveraging various fully connected recurrent neural networks with LSTM cells, our system promises robust fault detection across the board. With rigorous training and testing conducted within the dynamic confines of the Gazebo simulator, our method stands ready for real-world deployment. As further future directions, our sights are set on validating this approach on actual multi-rotor platforms. Furthermore, we envision extending its application to a myriad of multi-copter configurations, from hexarotors to quadrotors with plus configurations, unlocking a realm of possibilities for safer, more reliable aerial operations.

ACKNOWLEDGMENT

The research leading to these results has been supported by the COWBOT project (in the frame of the PRIN 2020 research program, grant number 2020NH7EAZ_002), the AIDROW project (in the frame of the PRIN 2022 research program, grant number 2022BYSBYX, funded by the European Union Next-Generation EU), and the European Union's Horizon programme under Grant 101070596 (euROBIN project). The authors are solely responsible for its content.

REFERENCES

[1] S. Zhao, F. Ruggiero, G. A. Fontanelli, V. Lippiello, Z. Zhu, and B. Siciliano, "Nonlinear model predictive control for the stabilization of a wheeled unmanned aerial vehicle on a pipe," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4314–4321, 2019.

[2] S. Roos-Hoefgeest, J. Cacace, V. Scognamiglio, I. Álvarez, R. C. González, F. Ruggiero, and V. Lippiello, "A vision-based approach for unmanned aerial vehicles to track industrial pipes for inspection tasks," in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1183–1190, 2023.

[3] S. D'Angelo, F. Pagano, F. Ruggiero, and V. Lippiello, "Development of a control framework to autonomously install clip bird diverters on high-voltage lines," in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 377–382, 2023.

[4] M. Izadi and R. Faieghi, "High-gain disturbance observer for robust trajectory tracking of quadrotors," *Control Engineering Practice*, vol. 145, p. 105854, 2024.

[5] R. Venkataraman, P. Bauer, P. Seiler, and B. Vanek, "Comparison of fault detection and isolation methods for a small unmanned aircraft," *Control Engineering Practice*, vol. 84, pp. 365–376, 2019.

[6] K. ichi Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural Networks*, vol. 6, no. 6, pp. 801–806, 1993.

[7] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[8] D. Guo, M. Zhong, H. Ji, Y. Liu, and R. Yang, "A hybrid feature model and deep learning based fault diagnosis for unmanned aerial vehicle sensors," *Neurocomputing*, vol. 319, pp. 155–163, 2018.

[9] I. Samy, I. Postlethwaite, and D.-W. Gu, "Sensor fault detection and accommodation using neural networks with application to a non-linear unmanned air vehicle model," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 224, no. 4, pp. 437–447, 2010.

[10] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018.

[11] X. Zhang, Z. Zhao, Z. Wang, and X. Wang, "Fault detection and identification method for quadcopter based on airframe vibration signals," *Sensors*, vol. 21, no. 2, 2021.

[12] V. Sadhu, S. Zonouz, and D. Pompili, "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.

[13] B. Wang, D. Liu, Y. Peng, and X. Peng, "Multivariate regression-based fault detection and recovery of uav flight data," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 3527–3537, 2020.

[14] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *RotorS - A Modular Gazebo MAV Simulator Framework*, vol. 625, pp. 595–625. 01 2016.

[15] A. De Luca, A. Albu-Schaffer, S. Haddadin, and G. Hirzinger, "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.

[16] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello, "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.

[17] F. Ruggiero, M. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Peréz, V. Lippiello, A. Ollero, and B. Siciliano, "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.

[18] F. Ruggiero, "Decentralized control of aerial manipulators through a momentum-based estimator," in *Aerial Robotic Manipulation. Research, Development and Applications* (A. Ollero and B. Siciliano, eds.), vol. 129, pp. 159–174, Cham: Springer, 2019.

[19] T. Madani and A. Benallegue, "Backstepping control for a quadrotor helicopter," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3255–3260, 2006.

[20] F. Ruggiero, M. A. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Peréz, V. Lippiello, A. Ollero, and B. Siciliano, "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.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[22] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, p. III–1310–III–1318, 2013.

[23] F. W. Vann, "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 (nonmembers).," *The Aeronautical Journal (1968)*, vol. 93, no. 930, p. 406–406, 1989.