IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 June 2024, accepted 2 August 2024, date of publication 9 August 2024, date of current version 19 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3441115

RESEARCH ARTICLE

Dynamic Flight Tracking: Designing System for Multirotor UAVs With Pixhawk Autopilot Data Verification

STANISLAV KUSMIREK[1], VLADIMIR SOCHA[1], (Senior Member, IEEE), TOMAS MALICH[1], LUBOS SOCHA[1,2], KAREL HYLMAR[1], AND LENKA HANAKOVA[1]

[1]Department of Air Transport, Czech Technical University in Prague, 128 00 Prague, Czech Republic
[2]Department of Air Transport Management, Technical University of Kosice, 041 21 Košice, Slovakia

Corresponding author: Vladimir Socha (sochavla@fd.cvut.cz)

This work was supported by the Internal Grant Agency of Czech Technical University in Prague under Project SGS23/136/OHK2/2T/16.

ABSTRACT The increasing diversity of multirotor unmanned aerial vehicles (UAV) designs poses significant challenges in evaluating their flight characteristics and performance parameters. This is particularly true for commercially available UAVs whose control units do not provide users with data that could be used, for example, to estimate the operational and flight limits of the flight device itself. This study introduces the concept of a mountable device designed to track and assess the flight parameters of quadcopters, independent of the UAV's systems. Specifically, it involves independent monitoring of rotations, flight speed, air pressure, temperature, and drone orientation in space. The device validation involved real flight tests with the IRIS+ quadcopter using the Pixhawk control system, whose data were taken as a benchmark for validation. To demonstrate the applicability and benefits of such a device, the study also created a concept of an operational envelope for the drone, i.e. dependence of thrust on weight, angle of attack and speed. This concept was created using robotic simulation in the Gazebo environment. In the simulations, the IRIS+ device was used to simulate flights with different payload weights (0.9–2.9 lb, approximately 0.4–1.3 kg) and twenty flight speeds (1–20 ms^{-1}), while simultaneously monitoring the angle of attack and motor revolutions (subsequently converted to thrust). The created operational envelope was subsequently validated using data from real flights. The overall results demonstrated the successful validation of the designed device and the accuracy of the measurement of critical flight parameters, with rotation measurement errors ranging between 100–200 RPM, angle of attack error at 4.25°, and altitude measurement error based on pressure at 0.56 m. Additionally, within this concept, the measurement of indicated airspeed was introduced, reflecting the expected flight speed values. In the context of validating the operational envelope, the results showed that the parameters of real flights fell within the predicted area of the created operational envelope for the IRIS+ drone. Independent monitoring devices like this can benefit operational limit determination and other testing purposes, especially for UAVs lacking data-sharing control units.

INDEX TERMS Aviation, flight tracking, quadrocopter, quadrocopter flight characteristics, robotic simulation, unmanned aerial vehicle.

I. INTRODUCTION

Knowledge of quadrocopters' technical and performance characteristics is an important prerequisite for determining their suitability for use and ensuring the required safety during their integration into common airspace in relation to the relevant existing legislation. The basic international regulatory framework, defined by the International Civil Aviation Organization (ICAO) since 2011 [1], addresses the technical and operational issues of using Unmanned

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Tang.

Aircraft Systems (UAS) through standards and recommended practices, along with supporting procedures for aeronautical and navigation services, as well as advisory materials for routine UAS operations globally. In this context, aviation organizations such as the European Union Aviation Safety Agency (EASA) and the Federal Aviation Administration (FAA) have issued regulations, including the Commission Delegated Regulation (EU) 2019/945 of 12 March 2019 on unmanned aircraft systems and third-country operators of unmanned aircraft systems, the Commission Implementing Regulation (EU) 2019/947 of 24 May 2019 on rules and procedures for the operation of unmanned aircraft, and the CFR Part 107 (The Code of Federal Regulations Part 107, [2]). The common objective of EASA and FAA is to create certification standards for UAS and establish processes for assessing their operational risks in managing unmanned transport. Integrating UAS as a new component of the aviation system into the national non-segregated airspace is generally managed and regulated in relation to valid international legislation by the national civil aviation authorities of individual countries.

According to a report by Market Reports [3], the global market size for unmanned aerial vehicle (UAV) transportation and logistics is estimated to have reached US$14.81 billion in 2021. The forecasted Compound Annual Growth Rate (CAGR) from 2022 to 2028 is expected to be 18.4 % per annum. A significant contributor to the increased total value of the transport and logistics market is the advent of 5G network technology and backhaul connectivity (the links between the DSLAM and the Internet Service Provider's IP) [4]. Investments in infrastructure reflect the market's confidence in the potential of UAS, thereby naturally elevating the demands for their reliability and safety.

Understanding quadrocopters' technical and performance characteristics is essential for assessing their suitability for use and guaranteeing the necessary safety during their integration into shared airspace. Currently, no comprehensive standards and testing procedures have been established specifically for UAS in the quadrocopter subcategory. Consequently, verification methods traditionally used in larger aviation sectors are often employed, although these do not encompass a thorough analysis of all essential parameters. Analysis of quadrocopters' aerodynamic and performance traits is typically conducted through software simulations [5], [6], [7], [8], experimental measurements in wind tunnels [9], [10], or actual flight tests using fully operational quadrocopters.

In the aviation industry, software simulations are grounded in complex mathematical analysis tools. These simulations are utilized primarily for evaluating the aerodynamic and performance characteristics of quadrocopters or their components, employing Computational Fluid Dynamics (CFD) [7]. They extend to simulating the operation of control regulators [11], robotic simulations focused on parallel flights [12], and the verification of anti-collision systems [13]. The key advantage of software testing over hardware testing is the ability to quickly and efficiently fine-tune various investigated configurations and parameters without necessitating the physical presence of the quadrocopter being tested. This significantly reduces the costs associated with testing. However, the main limitation of software testing is its inability to replicate real flight conditions fully. Consequently, the software testing results can only provide an approximation of the flight characteristics and performance of the quadcopter, which may vary from the actual results obtained in real-world measurements.

The wind tunnel testing method for quadcopters is utilized to evaluate their aerodynamic, performance, and efficiency flight characteristics [9], [10], [14]. Results from such tests include, for instance, the generation of thrust maps. These maps demonstrate the impact of airflow rate and the tilt angle of the quadcopter, or solely its propulsion units, on the total mechanical thrust. Additionally, they involve creating diagrams showing the mechanical efficiency of the motors as a function of airflow speed [14].

Testing performed through actual flights can be categorized into indoor flight tests [15] and outdoor flight tests [16]. In indoor environments, typical tests include accuracy assessments of maintaining the hovering position [17], flying according to a predetermined mission, parallel flying, or group flying. These tests are designed to verify the accuracy of the control system algorithms of quadrocopters and their intercommunication channels. Generally, in indoor environments, there are challenges with the GPS signal, which is essential for position maintenance, as well as the control loop of the controller. Consequently, during these tests, the satellite system is often replaced by an alternative positioning system, most frequently a Motion Capture camera system.

Quadrocopter flight testing in an outdoor environment becomes a crucial step in validating its flight and performance characteristics [16]. These dynamic tests in outdoor settings encompass various types, including stability, controllability, maneuverability, range, and endurance tests. The limitations of outdoor testing stem from the uncontrolled environment, which may include diverse wind gusts, and the methods used for data collection essential for evaluating the required flight characteristics. Flight data can be gathered using a control unit, a data logger capturing data in the form of MAVLink messages from the control unit, or a sensor network recording the necessary performance characteristics of the quadrocopter.

Traditional static or dynamic testing methods provide a comprehensive overview of the aerodynamic properties and performance of propulsion units in UAV. While software simulations are beneficial in the early stages of development due to their cost-effectiveness and the rapidity with which various modifications can be implemented, they are limited in accurately simulating real-flight conditions. Conversely, real dynamic flight tests, crucial in later development stages, face

limitations related to cost, the breadth of parameters tested, and the impact of weather conditions on result accuracy.

Each method of testing the flight or performance characteristics of UAVs has its own specifications and limitations. In the development of lower weight class UAS (e.g., C1), intended for the general public, or modifications of existing UAVs with various camera or functional systems, the financial cost of certain development and testing cycles, such as wind tunnel testing, is often a limitation. When considering substituting expensive prototype testing with real flights, it is necessary to ensure the availability of UAV flight and performance data. However, this requirement is often unmet when using low-cost control units or closed-system control units. This data deficiency can be addressed by adding a mountable sensor device to the tested UAV, which could accurately record or stream flight and performance data. These data can then be used to create the required operational and flight envelopes.

Therefore, the main objective of this study was to develop a modular device capable of recording the flight and performance parameters of a quadcopter and verifying its accuracy using flight data from the UAV autopilot system. Furthermore, flight characteristics from real flights captured by that device were compared with simulated flights using the envelope of the thrust dependence on the angle of attack and velocity of the UAV.

To achieve the goal of this study, the process of selecting the UAV and identifying the key parameters needed to validate the device is first outlined. The development and use of a redundant measurement system (DroneBox) are then detailed, along with the testing methods used in both simulated environments and real outdoor flights. Finally, the results are presented, showing how the flight parameters were verified and an operational envelope was established based on simulated data, which was then compared with real-flight data.

II. MATERIALS AND METHODS

Newly proposed methods were based on the indirect determination of UAV performance characteristics from flight data, either software-simulated flights or real outdoor flights in an outdoor environment. The basic parameter for mutual comparison of testing results was the type of tested UAV, the choice of which had to be in accordance with the specifications of the proposed testing methodologies.

A. CHARACTERISTICS OF THE IRIS+ QUADROCOPTER AND PIXHAWK AUTOPILOT

In selecting the appropriate quadrocopter model, key hardware specifications, including maximum take-off weight and performance requirements as determined by the UAV ability to fly with a minimum payload of 200 g, were considered. The software requirements emphasized the capability for autonomously controlled flights via a suitably chosen ground control station and the control unit's data architecture, which enable access to flight data through the MAVLink data stream.

TABLE 1. Technical specifications of used quadrocopter IRIS+.

Autopilot hardware	Pixhaxk 2.4.8
Firmware	Quadrotor Wide 1.14.0 dev
GPS	3DR uBlox GPS with Compass (LEA-6H module, 5 Hz update)
Telemetry radio	3DR Radio Telemetry V2 433 MHz
Motors	950 kV, MN2213-type
Propellers	3D robotics Solo 10×4.5" Self-Tightening Props, 2×CW, 2×CCW
Battery	3 cell 11.1 V 6 100 Ah lithium polymer
Flight weight with device	1.6 kg

In line with the specified criteria, a commercially available VTOL UAV of the multicopter type, specifically the IRIS+ model from 3DRobotics (California, USA), was chosen to assess the precision and functionality of the proposed measurement system in outdoor flight conditions. It is important to note that the mentioned type of UAV is utilized in various configurations (concerning motors, propellers, and batteries) and therefore it is not feasible to provide a specific datasheet for the particular quadrocopter model. The described characteristics represent the utilized model for the purposes of this study (see Table 1). The quadrocopter was equipped with four brushless DC motors of MN2213-type with a speed constant of 950 kV, arranged in an asymmetric configuration of X. Attached to each motor shaft was a self-tightening two-blade propeller of the 3D Robotics Solo type, with a diameter of 254 mm and a pitch of 114 mm (10'' × 4.5''). The quadrocopter's take-off weight without the battery reaches 1 kg. The manufacturer estimates the average flight time per one 11.1 V 3500 mAh lithium-polymer battery to be between 10 and 13 minutes, depending on the load, weather conditions, flight mode, and maneuvers performed. In order to increase the number of possible flights per battery, 11.1 V lithium-polymer batteries with a capacity of up to 6100 mAh were used during outdoor test flights.

Quadrocopter IRIS+ is outfitted with a 32-bit Pixhawk 2.4.8 control unit. This unit, integrated with QGroundControl firmware, enables the execution of autonomous missions. The control unit has an integrated barometer, magnetometer, 3-axis accelerometer, and gyroscope. A peripheral sensor, the uBlog GPS module with a compass, is connected to this unit. The system includes a telemetry radio set, ensuring bidirectional communication between the UAV and the ground control station or the operator's controller via the MAVLink data protocol. The control unit's internal memory allows for the recording and backup of all transmitted and generated data. Manual flight control is achieved through a 3DRobotics remote control, which, according to the manufacturer, has a range of up to 1 km [18]. However, under real flight conditions, the range of radio control is typically between 100 to 150 meters. The firmware of the control unit enables the programming of the UAV mission with precisely

defined flight parameters, such as flight altitude, velocity, seeding time over a point, or rotational conditions around the z-axis. The program is uploaded to the UAV's control unit via the telemetry system, and upon switching to the *mision flight* mode, it executes a fully autonomous flight.

B. DEVELOPMENT AND APPLICATION OF THE REDUNDANT MEASUREMENT SYSTEM DroneBox

A fundamental requirement of the measurement system was to provide users with a flight record containing time-synchronized data of key flight parameters. These testified about the linear and angular acceleration of the quadrocopter during flight, positional angles on all three axes, flight height, magnetic course, the velocity of the UAV relative to its environment (indicated airspeed), and the revolutions of each propulsion unit separately. The selected flight parameters for monitoring were chosen to adequately evaluate the quadrocopter's status for post-flight analysis of the UAV's performance in outdoor conditions. The design emphasized minimizing the sensor network's weight while enabling the recording device's application on Class C1 UAS.

The architecture of the system, designed for dynamic monitoring of the flight characteristics of UAV, is protected by patent *Device for Measuring Performance Parameters of Multicopters* (PV 2020-724). The main control element of the system, called DroneBox, is the Raspberry Pi Zero 2W microcontroller, complemented by two Arduino microcontrollers – Nano 33IoT and Micro (see Fig. 1). This configuration provided a versatile and robust platform for data preprocessing and collection. Power for the device comes from the UAV's internal source via a Castle programmable voltage converter regulator, lowering the original 11.1 V from the IRIS+ battery to the necessary 5 V. Choosing an internal power source positively impacted the system's overall weight and ensured a stable and suitable voltage for all components.

Sensor modules in the system were categorized into two groups based on their input voltage and signal processing requirements. Connected to the Arduino Micro microcontroller, the first category includes the MPXV7002DP total, static pressure analog converter, and an inertial measurement unit (IMU), specifically the GY-87 module. This module combines a three-axis gyro-accelerometer (MPU6050 chip), a three-axis magnetometer (HMC588L chip), and a pressure gauge (BMP180 chip). These sensors provide data on the UAV's indicated airspeed, linear and angular accelerations, magnetic heading, and atmospheric pressure. Utilizing available data, the module computes positional angles on all three axes. The second group of modules, connected to the Arduino Nano 33IoT, comprises sensors with an input voltage of up to 3.3 V, such as the 3144E magnetic sensors with a Hall probe. These modules are designed for measuring propeller revolutions by detecting magnetic fields, specifically through the number of magnetic field detections when a propeller blade with a fixed neodymium magnet passes by.

Dronebox's total system weight reached approximately 110 grams. To enhance the system's resistance to vibrations, fiber conductors were selected. The design solution aimed to minimize the aerodynamic impact on the UAV when applying the system. Consequently, system modules and the analog converter are strategically located under the cover in a dislocated manner, ensuring the system's weight minimally affects the UAV's balance. Considering the potential application of the DroneBox device on various types of quadrocopters or multicopters, the placement of microcontrollers and sensors was designed to be as universal as possible. For under-cover placement, it is advisable to align the GY-87 module parallel to the longitudinal x-axis and the transverse y-axis of the quadrocopter's control unit. Placing the IMU module in close proximity to or in contact with the UAV control unit is not recommended due to the potential mutual interference of magnetic compasses.

A suitable position for the magnetic modules with the Hall probe, used for the measurement of propeller revolutions, is on the arms of the quadrocopter, at least 10 cm away from the axis of the electric motor. Failing to maintain this distance may result in the Hall probe capturing magnetic vortices generated by the UAV's electric motor. For accurate measurements, attaching small neodymium magnets (up to 3 mm in diameter) to the lower part of both propeller blades is crucial. These magnets signal the propeller's half-turn to the magnetic sensor. The distance between the Hall probe and the magnet should not exceed 10 mm due to the cubic decrease in the intensity of the magnetic field. To ensure correct airspeed measurement when the pitot-static tube is not placed on the gimbal system of the camera, a modification in the seal's geometry or the tip of the tube is required [19]. The tube should be positioned on the UAV's undercarriage in a location minimally affected by air currents from the propellers.

The software solution for the DroneBox system was developed separately in two programming languages, C++ and Python, each used for different components. The Arduino Micro microcontroller utilized I2C communication with the GY-87 module and signal measurement via an analog 5V input pin; or digital interrupt inputs for all inputs from magnetic sensors with a Hall probe, handled by the Nano 33IoT microcontroller. Initially, communication between the Arduinos and the Raspberry Pi microcontroller was established through UART and SPI communication. However, this method proved inadequate under real load conditions, leading to the adoption of USB as the optimal serial communication channel for mutual data exchange. User-system communication was facilitated through the PuTTY application, once the Raspberry Pi's IP address was identified via the VNC Viewer. To initiate measurement, users executed a Python script, which began recording the measured data to the microcontroller's internal memory. The recording could be terminated either through a command in the controller's command window or by switching off the power source.

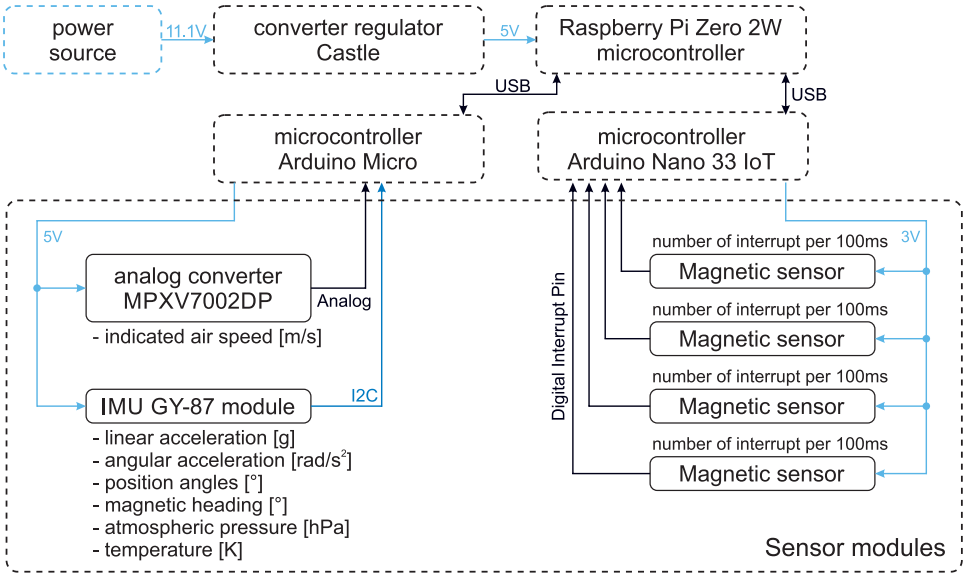

FIGURE 1. DroneBox onboard recording device architecture: displayed types of modules, measured values, information flow within the system, and power supply.

Final construction design of the DroneBox measurement device demonstrated no issues with vibrations that occur during actual UAV flights. This approach had a minimal impact on the aerodynamics of the test quadrocopter, causing only a slight increase in the overall weight of the UAV. Enhancing the user interface, the system was configured for initiation via the Raspberry Pi microcontroller and VNC Viewer, facilitating immediate error detection during measurement. However, a significant drawback of the DroneBox device, particularly when installed under the quadrocopter's cover, was its lengthy installation process on the UAV, necessitating the complete disassembly of the cover.

C. FLIGHT TESTING USING A 3D ROBOTIC SIMULATOR

The selection of UAV and its control unit, specifically the Pixhawk series autopilot, predetermined the simulation program's choice and its components. The Robot Operating System (ROS) Melodic was employed as a component of the meta-operational system for robotic flight simulation using the Software In The Loop (SITL) method. The ROS fully supports Python, C++, and Lisp programming languages [20], [21], [22]. Considering the compatibility of the robotic simulator's components, Gazebo (version 9.0.0) was selected as the physics and rendering simulation tool for flight testing. Another integral component of the robotic simulator was the Pixhawk autopilot (PX4), which supports SITL-type simulations and fully integrates with the ROS. In the root directory of the autopilot, configurations of individual basic UAS with corresponding control algorithms of specific types are available. The system allows for easy customization of control algorithms and facilitates adding, modifying, or importing sensors not directly connected to the UAV motherboard. The modularity of the PX4 combines hardware components with a software interface, enabling the autopilot to collect data from sensors at deterministic intervals through a control loop. Flight data about the state of UAV is recorded in the .ulog format.

To control the simulation, it was necessary to ensure seamless integration of the robotic simulator with the work environment component. The role of the working environment was served by the ground control station QGroundControl. The chosen Ground Control Station (GCS) was developed as part of the DroneCode project, just like the PX4 autopilot system, ensuring their full compatibility. The MAVLink (Micro Air Vehicle Link) communication protocol facilitated communication between the individual components of the robotic simulator.

During simulated flights in the Gazebo robotic environment, flight parameters were recorded for the IRIS+ quadrocopter at weights ranging from 0.9 lb to 2.9 lb, with intervals of 0.2 lb. Since Gazebo employs a basic linear aerodynamics model and the APC SF 10"$\times$ 4.7"propeller as the default for robotic simulations of the IRIS+ type UAV, modifications to the hardware configuration of the *iris.sdf* model was necessary for each simulation. In addition to the UAV's weight (in lb), the modified data also included the *motor* plugin's specific parameters: *motor_constant* and *moment_constant*. Changes in the values of *rolling_moment* and *rotor_drag_coefficient* parameters based on the H-force were neglected due to the unchanged diameter of the propeller.

According to the source code [23], the thrust of the motor is defined as

$$\text{force} = \text{real\_motor\_velocity}^2 \times \text{motor\_constant} \quad (1)$$

which means,

$$T = \omega^2 \times K \tag{2}$$

Converting to standard notation, *motor_constant* K can be calculated as

$$K = \frac{T}{(2\pi n)^2} \tag{3}$$

where T is thrust in N and n is the motor's revolutions per second in s^{-1} [24], [25]. The Gazebo model calculates the magnitude of rotor torque as

$$\text{rotor\_torque} = \text{force} \times \text{moment\_constant} \tag{4}$$

According to standard propeller theory, the moment constant C_M is expressed as

$$C_M = \frac{Q}{T} = \frac{C_Q \rho n^2 D^5}{C_T \rho n^2 D^4} \tag{5}$$

from which it follows

$$C_M = \frac{C_Q}{C_T} \times D \tag{6}$$

where C_Q is the moment coefficient, C_T the thrust coefficient, and D is the propeller diameter in m. The values of thrust and revolutions used to calculate the *motor_constant* and the coefficient values employed to calculate the *moment_constant* were sourced from a table in a study that focused on the static testing of propulsion elements for small multirotor UAV [26].

Flight testing was conducted in a precisely defined environment. The duration of the flight was not dependent on battery capacity. The maximum telemetry distance from the Ground Control Station (GCS), which was contingent on the transmitter's signal, was determined to be within a radius of 375 meters.

Nine autonomous flights were simulated with different UAV weights and a consistent trajectory. Each flight began with ascending to a height of 10 meters, followed by a hovering maneuver lasting 30 seconds. The flight then continued with a maneuver of horizontal forward straight-line flight, ranging in velocity from 1 $\text{m} \cdot \text{s}^{-1}$ to 20 $\text{m} \cdot \text{s}^{-1}$. Each horizontal flight was concluded with a 2-second hover and a 180° turn in preparation for the next horizontal flight at increased speed. After completing all the flights, the quadrocopter returned to the original take-off point and landed (see Fig. 2-A). The total length of one simulated flight was approximately 9650 m. The entire flight was monitored on the GCS QGroundControl screen and the Gazebo graphic simulator window.

After the simulated flight was completed, the flight data was retrieved from the GCS in .ulg format and converted to .csv format as part of each MAVLink quadcopter autopilot message. For further evaluation of the messages, the orientation of the UAV in space was calculated from the quaternions found in the *vehicle attitude setpoint* message. Flight altitude and ground speed were acquired from *vehicle GPS position* GPS messages; barometric height and air density from *vehicle air data* reports; and the speeds and performance of each engine were extracted separately from the *esc status* messages.

D. OUTDOOR FLIGHTS AND DATA COLLECTION

The recording of flight data during test flights was conducted in a duplicate manner. It took place on the control unit of the quadrocopter (Pixhawk 2.4.8 autopilot) and the DroneBox device, which was integrated into the UAV. Real flights were carried out over the runway of model airfield LMK Velka Lomnica, Slovakia. Flight activities were reported to the flight information service. The flight tests were conducted under wind conditions up to 5 $\text{m} \cdot \text{s}^{-1}$, with the wind continuously measured by a UNI-T UT363S anemometer positioned near the GCS. The maximum telemetry distance from the GCS was limited to 100 meters.

In total, 94 autonomous flights with identical trajectories were completed, each covering approximately 1600 meters. Pre-flight preparation involved calibrating sensors in the quadrocopter's control unit, which was performed using the GCS QGroundControl software. The UAV was then positioned on the starting spot on the runway, ensuring that the front orientation of the quadrocopter was directed towards the first point of the pre-programmed flight mission trajectory. Remote access to the DroneBox device was established via WiFi communication, with flight record logging initiated through the VCN Viewer interface. The next step involved checking the position of the control sticks on the IRIS+ controller, which was responsible for selecting the flight mode. Confirming that the sticks were in the ''manual flight'' and ''altitude flight'' positions, the controller was switched on. After verifying the connection between the UAV, GCS, and controller, the controller was switched to ''mission flight'' mode, commencing the autonomous flight.

The test flight commenced with the quadrocopter ascending to an altitude of 10 meters, followed by a hovering maneuver for 3 seconds. Horizontal forward maneuvers followed at velocity according to the program prepared for autonomous flight. Upon reaching the final position, the quadrocopter remained in hover for 2 seconds, then rotated 180° and continued with the mission. The range of measured velocity was from 5 to 14 $\text{m} \cdot \text{s}^{-1}$. The velocity range was determined due to the limited range of radio control, which directly influenced the flight time at the required steady speed. Flight plan of one outdoor flight is shown in Fig. 2-B. From an operational safety standpoint, limiting the GCS range to 100 meters allowed the operator to intervene in control if necessary (beyond 100 meters, it is impossible to discern the front part of symmetric quadrocopters). The flight was monitored on the QGroundControl GCS window by the operator, who was ready to intervene in-flight anomalies (unauthorized trajectory changes, altitude variations, etc.). During all measured flights, the operator intervened in the control five times.

After completing all maneuvers within the autonomous flight program, the quadrocopter returned to the starting point

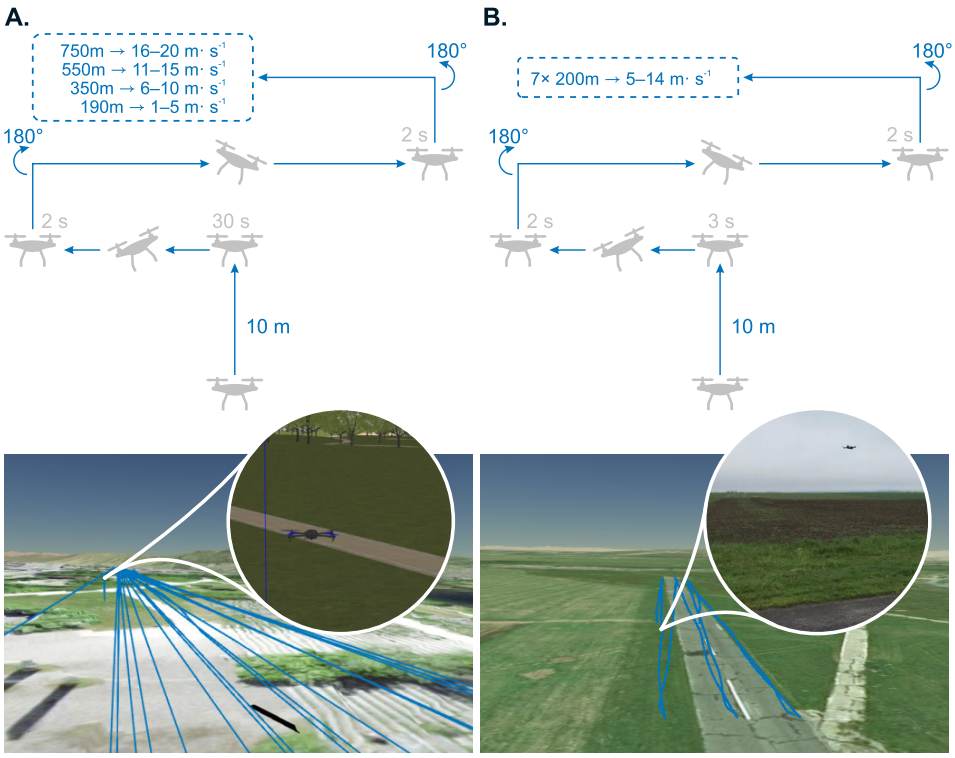

FIGURE 2. Scheme of maneuvers performed within one flight at a flight altitude of 10 m in robotically simulated flights (A) and outdoor flights (B), along with examples of flight trajectories for both setups.

and landed. Following landing and the engine disarmed, the mode on the controller was switched to "manual flight", and the transmitter turned off. The data recording by the DroneBox device was stopped via remote access or by turning off the UAV's power supply (battery). A change in the control unit's program, defining the flight velocity for the next flight, was made if the battery capacity was higher than 45 %. A maximum of 3 flights could be performed on a single battery charge.

Each flight day concluded with downloading flight data in .ulg format from the ground control station, which was then converted to .csv format as part of each MAVLink quadcopter autopilot message. The flight data was also retrieved directly from the DroneBox's internal memory in .csv format. For further analysis, the orientation of the UAV in space was calculated from the quaternions found in the *vehicle attitude setpoint* message as well thrust coefficient of all propulsion units. Flight altitude and ground speed were determined from the *vehicle GPS position* messages. Barometric altitude and air density were derived from the *vehicle air data* reports. The DroneBox recording provided detailed information on linear and angular acceleration in each axis, magnetometric data, barometric data, airspeed values, and tachometric values of each engine individually.

III. RESULTS

Systematic division of the results into several sections allowed for an in-depth examination of various aspects of the phenomenon under study. Essential outcomes included verifying the proposed DroneBox measurement system and creating an operational envelope for the UAV. This envelope reflects critical flight and performance characteristics such as thrust, angle of attack, and flight speed, particularly concerning the weight of the quadrocopter, which is crucial for enabling hovering and maintaining horizontal flight. Flight data processing enabled the creation and detailed analysis of this operational envelope. Comparative examination of the positions of different flight states within the operational envelope offered a deeper understanding of the UAV's performance parameters.

A. VALIDATION OF DroneBox DEVICE

Validation of the DroneBox device was undertaken using real flight data acquired from the Pixhawk autopilot. The graphically presented results illustrate one shortened autonomous flight, during which the control unit had to maintain the value of the velocity of the quadrocopter at 5 $\mathrm{m \cdot s^{-1}}$. This specifically selected flight is without turning the UAV at the end of the flight path (the UAV thus flew consistently on the front and back side of the quadrocopter in relation to the direction of flight).

The applied methodology of data preparation and synchronization for each evaluated parameter was uniform. A detailed explanation of this methodology is demonstrated in the example of altitude data evaluation, which provides

a model for the processing and analysis of other measured parameters.

In comparing altitude data from PX4 and DroneBox sensors, a methodology involving resampling and interpolating the data provided by DroneBox was employed. This process was followed by comparing altitudes recorded by both devices. It specifically entailed the computation of the mean difference between the altitudes given by PX4 and DroneBox, the calculation of the standard deviation of these differences, and determining the correlation coefficient.

Considering the different sampling frequencies of the provided time series, aligning these records in time was the first step. For this purpose, interpolation was used, creating a common time vector t_s that includes the time series from PX4 (t_{PX4}) and DroneBox (t_{DB}), i.e.:

$$t_c = \{t_{min}, \ldots, t_{max}\} \tag{7}$$

where $t_{min} = min(min(t_{PX4}), min(t_{DB}))$ and $t_{max} = max(max(t_{PX4}), max(t_{DB}))$. Following this step was the actual interpolation:

$$A_{PX4,int}(t) = interp(t, t_{PX4}, A_{PX4}) \tag{8}$$

and

$$A_{DB,int}(t) = interp(t, t_{DB}, A_{DB}) \tag{9}$$

where $t \in t_c$ and $interp(t, t_Z, A)$ denotes the interpolated altitude at time t based on the time series provided by device Z, which can be interpreted as (t_Z, A).

For the calculation of the actual "similarity" metrics, the computation was based on the residuals, i.e.:

$$D(t) = A_{PX4,int}(t) - A_{DB,int}(t) \tag{10}$$

from which the mean deviation (11), the standard deviation of differences (12), and the correlation coefficient (13) were calculated in the following way:

$$\mu_D = \frac{1}{N}\sum_{t\in t_c} D(t), \tag{11}$$

$$\sigma_D = \sqrt{\frac{1}{N-1}\sum_{t\in t_c}(D(t)-\mu_D)^2}, \tag{12}$$

$$\rho = \frac{\sum_{t\in t_c}(A_{PX4,int}(t)-\mu_{PX4})(A_{DB,int}(t)-\mu_{DB})}{\sqrt{\sum_{t\in t_c}(A_{PX4,in}(t)-\mu_{PX4})^2 \sum_{t\in t_c}(A_{DB,int}(t)-\mu_{DB})^2}}, \tag{13}$$

where μ_{PX4} and μ_{DB} represent mean values $A_{PX4,int}(t)$ and $A_{DB,int}(t)$.

Visualization of an example of the recorded altitude curves during the test flight is shown in Fig. 3-A. The mean difference between the altitude data is approximately 0.683 meters, with the PX4 sensor generally providing higher values than the DroneBox.

Based on the presented data, the drop in accuracy cannot be attributed to either the DroneBox device or the PX4. In both cases, temperature changes acting on the sensors can affect the measurement accuracy. The BMP180 sensor, as an integral part of the GY-87 module, is attached directly to the main frame of the kite above the battery of the quadrocopter. On the other hand, the MS5611 barometric sensor, which is part of the PX4 PCB, is covered by a second cover near the microprocessor of the control unit. An increase in the temperature of the quadrocopter during operation and, consequently, of the sensor can cause an increase in pressure, with the subsequent recording of a lower altitude. Of course, other quantities are included in the altitude calculation, but these can be approximated, or they are constants. From the barometric equation [27] expressed for the height h

$$h = \frac{T_0}{L}\left[1 - \left(\frac{P}{P_0}\right)^{\frac{RL}{gM}}\right] \tag{14}$$

it is obvious that the calculation depends on absolute temperature, the reference level T_0, and the temperature gradient L. The value of L is not universal and can vary depending on geographic location, time of year, and current meteorological conditions. However, in the standard atmospheric model, a linear temperature gradient is assumed. As with the previously discussed temperature, pressure measurement P can also introduce inaccuracies in altitude calculation, depending on sensor equipment. The values of reference pressure P_0 (at sea level), gravitational acceleration g, molar density of air M, and universal gas constant R are either constant, tabulated, or approximable.

The standard deviation of the altitude differences between PX4 and DroneBox is approximately 0.561 meters, indicating variability in the differences between these devices' recorded/calculated data. The correlation coefficient is approximately 0.985, which indicates a very high degree of linear dependence between the altitude data of both devices.

These metrics indicate that although data from both devices are highly correlated, there is a consistent, albeit small, difference in altitude readings, with the PX4 showing slightly higher values on average. After a thorough literature search, it was impossible to discover the exact procedure or formula by which the PX4 device calculates the altitude values. Furthermore, information about any potential temperature compensation of the input data that could affect the accuracy of the results is not available.

To understand the differences in altitude, it was necessary to analyze deviations between other directly measured variables of both devices. The analysis first focused on the measured atmospheric pressure. The mean difference between the measured atmospheric pressure values is approximately −0.079 Pa, indicating a slight deviation from both devices. This suggests that, on average, the PX4 sensor data are slightly lower than the DroneBox (DB) sensor data. The standard deviation of the differences is approximately 0.066 Pa, indicating relatively low variability between the devices. The correlation coefficient 0.983 suggests a very high degree of linear correlation between the atmospheric pressure data from both devices. Fig. 3-B visually illustrates

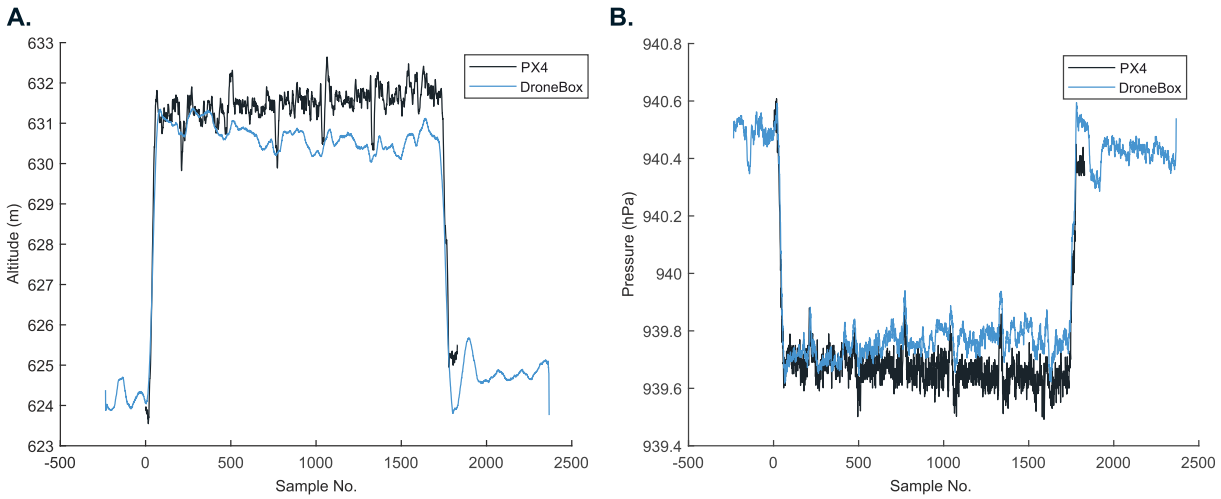

FIGURE 3. Validation of the altitude (A) and atmospheric pressure (B) parameters detected by the DroneBox device, as compared to Pixhawk (PX4) autopilot data.

these findings, displaying atmospheric pressure data from both devices. The high correlation coefficient, low mean difference, and standard deviation of differences indicate that both sensors provide similar and consistent atmospheric pressure readings.

The second source of deviation may lie in the measured temperature values. Analysis of temperature data from the PX4 and DB sensors revealed interesting findings. The mean difference between the temperature readings is approximately 5.53 °C, indicating a significant deviation in values from these two sensors. This suggests that the PX4 sensor's average temperature values are higher than the DB sensor's. Furthermore, the standard deviation of differences is approximately 3.00 °C, indicating a moderate variability in the differences between the temperature readings of both sensors.

Regarding the correlation coefficient, it is approximately 0.035, indicating a very low degree of linear correlation between the temperature readings from both sensors. This low correlation coefficient, along with the relatively high mean difference and standard deviation, suggests that the sensors have significantly different temperature readings and do not accurately capture mutual temperature changes. Visualization of these findings through Fig. 4 shows the temperature recording of both sensors in relation to the recorded sample.

Like the previous altitude data, the pitch data were preprocessed and evaluated using similar metrics. The mean difference between the pitch angle data is approximately 0.735°, indicating a minor deviation in data from both devices. The standard deviation of the differences is approximately 4.246°, suggesting a moderate variability in the differences between the pitch angle data from both sensors. The correlation coefficient is approximately 0.697, indicating a medium degree of linear correlation between the pitch angle data from both sensory devices. An example of the visual

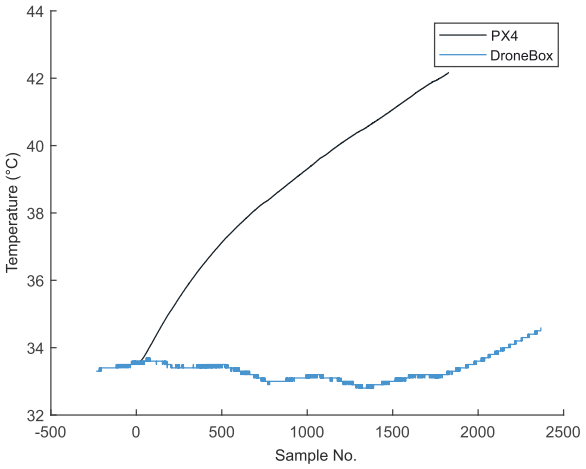

FIGURE 4. Validation of the temperature parameter sensed by the DroneBox device according to Pixhawk (PX4) autopilot data.

representation of the inclination angle values is shown in Fig. 5-A. Although there is some correlation, variability, and differences in the pitch values, and compared to the altitude data, the error in pitch angle is relatively higher, the result can still be considered satisfactory. However, it is necessary to consider possible factors contributing to this error. One possible explanation could be using sensors with different calibrations of zero angle, which can lead to slight deviations in measured values. Additionally, the impact of vibrations, especially if the PX4 autopilot is used on an anti-vibration platform, can affect the accuracy of measurements. In the context of error, it is also appropriate to consider the technical limitations of the DroneBox device, specifically the limited sampling frequency, which at low values may restrict the ability to capture rapid changes in inclination angle.

Fig. 5-B presents the ground speed measurements using the GPS module of the Pixhawk autopilot and the indicated

airspeed, recorded by the modified pitot-static tube of the DroneBox device. From the figure is clear that the indicated airspeed fluctuated between two significantly different values: approximately 5 $\mathrm{m \cdot s^{-1}}$ and less than 2 $\mathrm{m \cdot s^{-1}}$. These differences in values are noticeable during flight, with the front of the quadrocopter facing forward, when air flows into the pitot-static tube, allowing for accurate airspeed measurement. Conversely, in the case of backward flight, when the tube is oriented with the rear of the quadrocopter facing the direction of flight, the measurement of indicated airspeed was not relevant, as direct airflow into the tube was restricted. The measurement performed in this way points to the necessity of the correct orientation of the quadrocopter for reliable measurement of the indicated airspeed. The influence of wind gusts probably causes recorded oscillations in individual phases of flight in the correct orientation.

In the context of a comprehensive evaluation of the functionality of the DroneBox device, the analysis of individual quadrocopter propellers' revolutions remains a key validation element, as illustrated in Fig. 6. This parameter is crucial for assessing the performance attributes of the quadrocopter in real outdoor flight conditions, providing important information about the dynamics of the propulsion unit and its interaction with the surrounding environment.

Before interpreting these results, it is important to describe the approach to data processing. This is necessary because the PX4 does not provide revolutions information in the same way that the DroneBox does not provide, for example, thrust information. Therefore, for this type of evaluation, the rotations on the PX4 unit had to be approximated. The PX4 unit regulates rotations indirectly (i.e., the rotations are not measured directly), through pulse-width modulation (PWM) via the electronic speed controller (ESC), determining the speed of each motor unit. This allows control of the thrust generated by its rotor. To determine the proportion of thrust generated by all rotors, the thrust coefficient (Tc) is used in this context, derived from the knowledge of maximum thrust at maximum electrical voltage. Tc is then determined as the ratio of the voltage on all motors to the maximum voltage, thus taking values from 0 to 1. Tc essentially represents the total maximum thrust as a percentage. This parameter can then be used to calculate the thrust generated by individual rotors. However, this requires knowledge of the distribution of voltages across the individual motors, which was unavailable. The PX4 unit, however, directly provides information about PWM, which can be used to calculate thrust. Specifically, the calculation of thrust for individual rotors was based on the following equation:

$$T_i = T_{max} \frac{(PWM_i - PWM_{MIN})}{(PWM_{MAX} - PWM_{MIN})}, \tag{15}$$

where T_i is the resultant thrust generated by the ith propeller, PWM_i is the pulse width in μs on the ith motor, and PWM_{MIN} and PWM_{MAX} represent the minimum and maximum values of the pulse width (in this case 900 μs and 2000 μs). The fraction in the relation above represents the normalization of PWM_i to the interval 0…1. The values of PWM_{MAX}, PWM_{MIN} and T_{max} can be determined experimentally. From experimental static testing of the propulsion unit on the BenchMark RC Series 1580 test stand the relationship between the thrust of the 3D Robotics Solo propeller and the motor speed was determined to be expressed by a second-degree polynomial function,

$$T_i = p_1^2 rev_i^2 + p_2 rev_i + p_3, \tag{16}$$

where T_i is the thrust of the propeller at motor speed rev_i and the coefficients are denoted by p. The resulting revolutions on one rotor rev_i were therefore approximately equal to

$$rev_i = \frac{\sqrt{4p_1(T_i - p_3) + p_2^2}) - p_2)}{(2 \cdot p_1)}, \tag{17}$$

where the coefficients p were derived in the case of specifically used engines and propellers as follows: $p_1 \approx 1.049 \times 10^{-8}$, $p_2 \approx 4.044 \times 10^{-7}$ and $p_3 \approx -5.806 \times 10^3$. The values calculated in this way were subsequently compared with the actual rpm values read from the DroneBox device.

Propeller1 shows a relatively small mean difference (48.51 RPM) and standard deviation of differences (232.08 RPM). A high correlation coefficient (0.91) indicates a strong linear dependence between the revolutions recorded by PX4 and DroneBox. For propeller 2, values could not be captured via the DroneBox system during this sample measurement. Propeller 3 exhibits a higher mean difference (103.94 RPM) than propeller 1, but with a lower standard deviation of differences (162.87 RPM). The correlation coefficient (0.93) is also high, indicating a strong linear dependence, similar to propeller 1. Propeller 4 presents the largest mean difference (176.08 RPM) among the three monitored, with a medium standard deviation of differences (186.98 RPM) and a correlation coefficient of 0.85.

The results suggest that although PX4 and DB sensors tend to follow similar trends in revolutions (as evidenced by high correlation coefficients), there are noticeable differences in revolution measurements across different propellers. Mean differences and standard deviations of differences vary, indicating that these discrepancies are inconsistent across all propellers. This variability could be due to sensor calibration, disruptive magnetic interferences, environmental conditions, or operational differences between the propellers. In the case of PX4, the rotations were approximated from PWM. In such cases, it is possible that an error was already introduced into the expression of revolutions for PX4, for example, by linearizing the dependency of thrust on PWM. It should be noted that most control units, especially their software drivers, such as in the case of PX4 (Pixhawk 2.4.8), do not record revolution values. For DroneBox, the disadvantage is the inability to record or approximate thrust, for example, from knowledge of the voltage on the motors, etc. On the other hand, this device provides information about revolution, from which thrust can be approximated. It is thus a compromise,

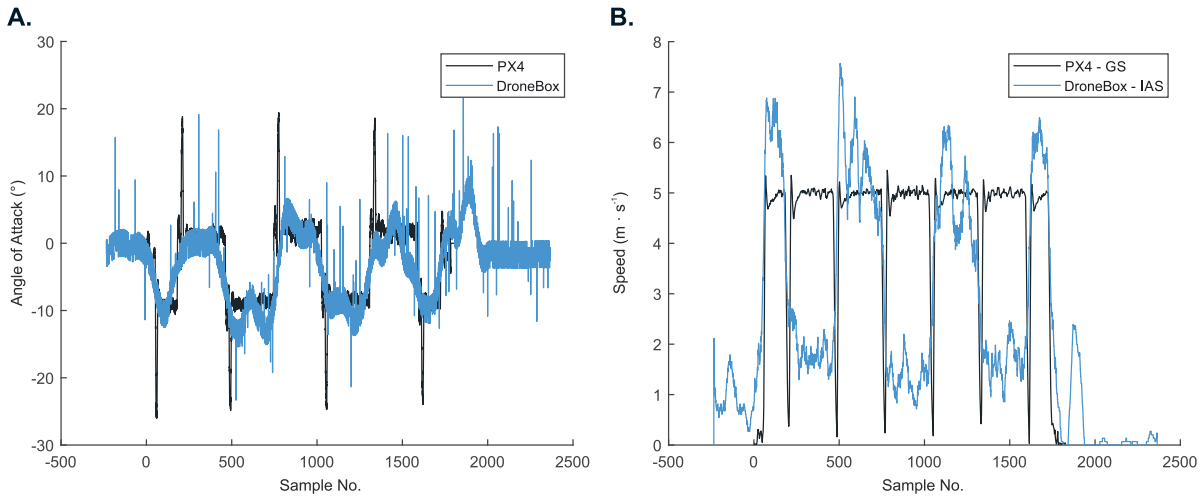

FIGURE 5. Validation of the angle of attack (pitch) parameter sensed by the DroneBox device according to Pixhawk (PX4) autopilot data (A), and comparison of ground speed (GS) based on PX4 autopilot data and indicated airspeed (IAS) in horizontal straight flight at 5 $m \cdot s^{-1}$ (B).

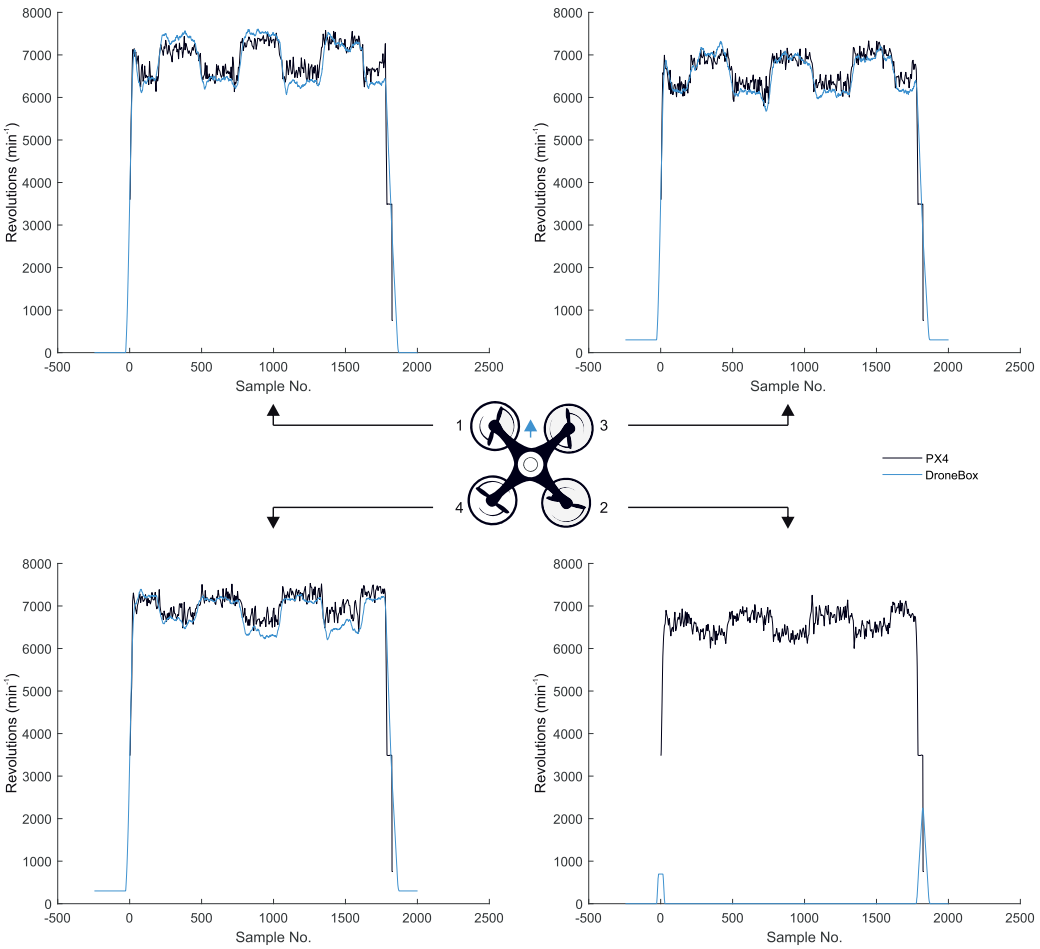

FIGURE 6. Comparison of recording of engine revolutions 1 to 4 of the quadrocopter according to the added data of the Pixhawk autopilot (PX4) and directly measured by the Hall probe of the DroneBox system.

in the context of which the results are relatively satisfactory, even though the cumulative variability in measurement seems to range between approximately 100–200 RPM and with absolute errors in the range of approximately 0–170 RPM.

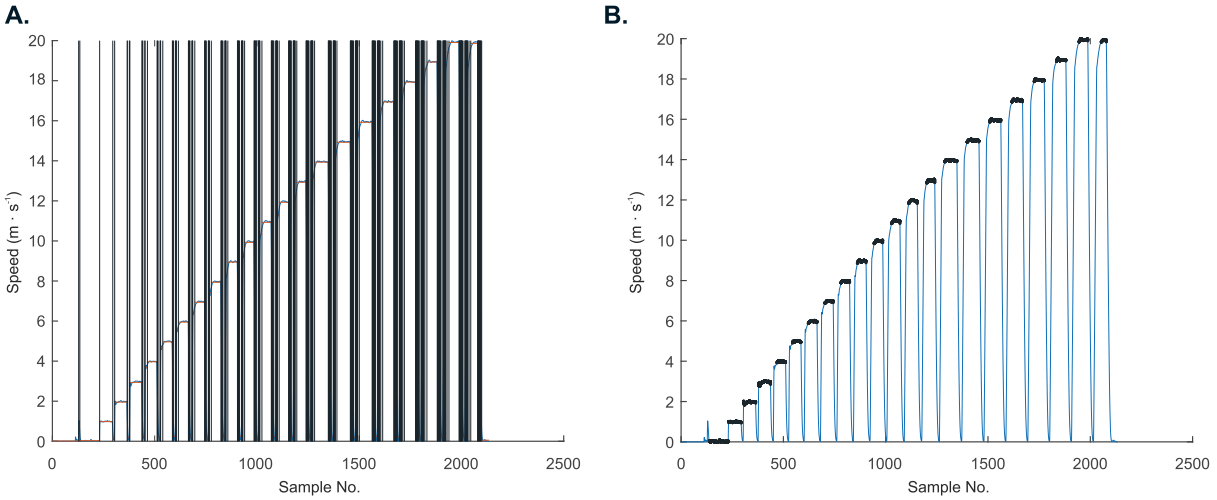

FIGURE 7. Identification of characteristic flight sections according to steady speed with a representation of all found sections using an algorithm designed for searching for sudden changes in the signal (A) and display of the resulting, identified flight sections (B).

B. OPERATING ENVELOPE

To construct the operating envelope simulated data were used. Before working with the data, it was initially necessary to divide the data from simulated flights into characteristic sections. A constant flight speed characterizes these sections. This process consisted of three parts, i.e., identification of characteristic sections in the data (Fig. 7-A), selecting sections with characteristic velocity (Fig. 7-B), and subsequent modification of these subsets. By defining the total velocity record in a single flight as ϵ, it is then composed of ξ sections with characteristic velocity, which need to be identified. The first step was to identify those parts of the flight that are characterized by their altitude. This can be done using different criteria.

In this article, an approach based on identifying sudden changes in the signal was adopted [28], [29]. This algorithm enables the detection of abrupt changes in the signal based on the change in residual error during a shift in linear trend, mean, standard deviation, or mean squared error. Specifically, the approach utilized involves monitoring the change in the linear trend in the data. For these purposes, a differential $d\epsilon$ was created from the velocity vector ϵ, defined as

$$d\epsilon_i = \epsilon_{i+1} - \epsilon_i, \text{ where } i = 1 \dots n-1. \tag{18}$$

The vector $d\epsilon$ was subsequently divided into τ parts, some of which defined the searched sections of the flight, see Fig. 7-A. The decision on which τ represented the desired ξ was made based on the knowledge of the length of the time series, specifically τ with a vector length of >270 data points (45 seconds). From the individual ξ, extreme values were removed, i.e., those values that do not fall within the natural distribution of the data. Since this process could have also eliminated values that caused discontinuities in the time series, additional linear interpolation was performed on the data. The resulting detected segments are then displayed in Fig. 7-B.

In this manner, a total of 9 simulated flights were divided, which are characteristic for the overall weight of the drone, specifically for 0.9 lb ($\approx$ 0.404 kg), 1.1 lb ($\approx$ 0.487 kg), 1.3 lb ($\approx$ 0.570 kg), 1.5 lb ($\approx$ 0.659 kg), 1.7 lb ($\approx$ 0.743 kg), 1.9 lb ($\approx$ 0.831 kg), 2.1 lb ($\approx$ 0.916 kg), 2.5 lb ($\approx$ 1.094 kg) and 2.9 lb ($\approx$ 1.314 kg). These weights will be an important factor in deriving the operating envelope of the UAV. Anyway, dividing these datasets resulted in D subsets defined according to the mass m and the flight speed v, and thus $D_{m,v}$, while the velocities were simulated in the range $1 \dots 20\ \mathrm{m \cdot s^{-1}}$. A total of 194 of these sub-datasets were created and the data contained in them describe the horizontal straight flight of the UAV.

Subsequently, these subdatasets were processed iteratively in the Matlab environment. The purpose of this processing was to find the dependency of thrust on velocity and angle of attack for each weight category. The visualization of these data is shown in Fig. 8, and the goal was to find a mathematical expression that would approximate the lines of this graph in 3D space. For these purposes, all datasets of a single weight, i.e., D_m, were sequentially loaded, and an approximation of these data was performed at two levels. The first level was fitting thrust to velocity. Given that the dependency was nonlinear in all cases, several numerical methods for fitting these curves were tested, from multi-order polynomial approximation through exponential fitting to power-law fitting. The best results and essentially the most intuitive interpretation were provided by fitting using a two-term power series, i.e.

$$T_m = c_1 \cdot v_m^{c_2} + c_3, \tag{19}$$

where T is the thrust and v is the flight speed for a given mass m. The coefficients $c_1 \dots c_3$ were subsequently calculated from the given notation. The second level was the pitch approximation using velocity. In this case, it appeared to be a linear dependence, and also for the sake of simplicity, this

TABLE 2. Thrust and angle of attack fitting results as a function of velocity.

	Thrust approximation				Angle of attack approximation			
m (kg)	SSE	R^2	df	RMSE	SSE	R^2	df	RMSE
0.404	0.197	0.996	8791	0.004	2994	0.996	8792	0.583
0.487	0.267	0.996	9440	0.005	3461	0.996	9441	0.605
0.570	0.296	0.996	9431	0.005	3651	0.995	9432	0.622
0.659	0.680	0.993	9415	0.008	5313	0.992	9416	0.751
0.743	0.790	0.993	9743	0.009	5433	0.992	9744	0.746
0.831	0.909	0.993	9720	0.009	5240	0.991	9721	0.734
0.916	0.813	0.994	9950	0.009	5590	0.990	9951	0.749
1.094	0.527	0.995	10258	0.007	3997	0.991	10259	0.624
1.314	1.464	0.912	6237	0.015	2176	0.977	6238	0.590

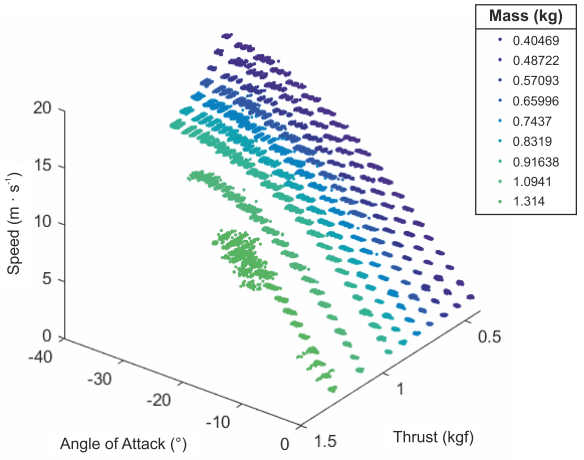

FIGURE 8. Visualization of thrust dependence on angle of attack and velocity for different simulated weights.

approach was used. Therefore,

$$N_m = c_4 \cdot v_m + c_5, \tag{20}$$

where N represents angle of attack for a given mass m and c_4 and c_5 are the coefficients sought.

The quality of each approximation is expressed using the sum of squares of deviations (SSE), the coefficient of determination R^2, and the square root of the mean of the squared differences between the approximated and actual values (RMSE) for the degrees of freedom (df) characterizing the number of data points forming each curve dependent on mass m. These data are provided in Table 2. In this context, SSE is a measure used to quantify the variation or deviation of a set of values from the actual observed values. The SSE is calculated by summing the squares of the differences between the observed (actual) values and the corresponding predicted values. This measure is commonly used in regression analysis to assess the fit of a model to data. A lower SSE value indicates a better fit, meaning the predicted values are closer to the actual observed values. The coefficient of determination, R^2, is a statistical measure used in the context of regression analysis to quantify how well a regression model explains the variability of the dependent variable. RMSE, as implied by the above description, thus provides a measure of the average magnitude of errors in model predictions, giving greater mass to larger errors (due to squaring the differences). Based on the information provided in Table 2, it can be stated that the fitted values exhibit relatively small error compared to the real values.

Another assumption was that the found coefficients would depend on mass and it would be possible to approximate them. This would make it possible to express the course of thrust and angle of attack depending on the assumed velocity and the determined weight. In other words, the creation of the envelope would be user-dependent only on the input parameters of velocity and mass. Therefore, the set of coefficients $c_1 \ldots c_5$ expressed for individual weights was subjected to analysis, the result of which was the subsequent approximation of these coefficients,

$$c_{a1} = 1.4e-5 \cdot e^{0.0085 \cdot m}, \tag{21}$$

$$c_{a2} = 3.205 - 0.0019 \cdot m, \tag{22}$$

$$c_{a3} = 0.02 + 0.0044 \cdot m, \tag{23}$$

$$c_{a4} = 0.0035 - 2 \cdot m, \tag{24}$$

and

$$c_{a5} = 0. \tag{25}$$

Thus, the above uses a simple exponential model for the c_1 approximation and a linear model in the remaining cases. The coefficient c_5 was not taken into account due to its insignificant influence on the fitting results and also to prevent the displacement of the curve on the y-axis.

The result of this processing is thus the possibility of estimating the angle of attack and thrust for a defined flight speed and weight in an assumed horizontal straight flight. For the example of defining the velocity vector $v = 0.001, 0.002, \ldots, 30\ \mathrm{m \cdot s^{-1}}$ and the mass $m = 90, 100, \ldots, 1000$ lb will be thrust defined as

$$T_{m,v} = 1.4e-5 \cdot e^{0.0085 \cdot m_i \cdot v_j^{3.205-0.0019 \cdot m_i}} + 0.02 + 0.0044 \cdot m_i, \tag{26}$$

and

$$N_{m,v} = 0.0035 - 2 \cdot m_i \cdot v_j + 0 \tag{27}$$

where i is the index of mass values in the vector m and j is the index of velocity values in the vector v. This particular case is presented in Fig. 9. This figure plots the values for thrust values less than 3.09 kgf, which is the maximum achievable

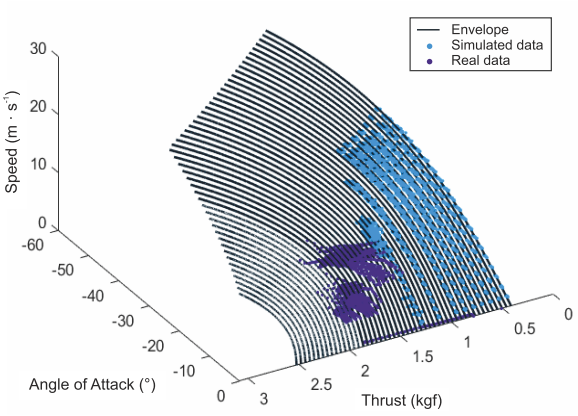

FIGURE 9. Visualization of the resulting envelope for a velocity of $<30\ \mathrm{m \cdot s^{-1}}$ and mass in the range of 90 to 1000 lb.

thrust of a real UAV. The presented graph also shows the horizontal parts of flights realized in a real environment. The real data plotted in the graph come from measurements for velocities 5, 6, 10, and 11 $\mathrm{m \cdot s^{-1}}$.

The lines in the presented graph represent ideal flight parameters for maintaining forward horizontal flight for the total weight of the UAV. The example of real data plotted on the envelope forms scattered clusters of data located below/above the established envelope; however, they provide a sufficiently good representation of the functionality of this solution. The weight of the UAV used for real flights is 1.6 kg (3.52 lb). The graph shows that flight sustained in forward horizontal flight falls within the range that applies to this weight. Zero values of angle of attack (pitch) and velocity on the thrust axis represent a hovering state and define the mass curve's starting point. Deviating from the ideal ''mass'' curve to lower thrust values will cause the UAV to descend. The operational area, which is bounded by the mass curve, is towards higher thrust values. This means that the UAV can operate at higher thrusts with a compromise on the angle of attack.

Several factors can cause the dispersion of data for real flights. One of the main ones could be noise, which is also visible in the data presentation. In addition, flights conducted in real environments can be influenced by meteorological factors, such as wind speed and direction. Although the envelope is constructed for ideal meteorological conditions, it defines the minimum requirements for conducting a flight while maintaining constant altitude.

IV. DISCUSSION

In the context of dynamic testing of UAVs under real operational conditions, a critical need arises for precise monitoring of their flight and performance characteristics. However, accessing these data in UAS is not a straightforward process. The availability of flight data depends on several factors, including the type of control unit (autopilot), ground control station, and the software firmware used.

Many low-cost UAS in the C1 category, which has limited autonomous flight capabilities, flight data are not systematically recorded but are used only in real-time. The situation is more complicated with commercial UAV, for example DJI, where access to raw flight data for scientific research is restricted, hindering the implementation of new control algorithms or pre-programmed changes in settings. Control units like ArduPilot or Pixhawk, on the other hand, are data-open and allow communication via MAVLink messages, making them popular for investigating UAV control processes in real flights. However, a potential problem is that some flight parameters are not automatically saved to the control unit's memory card without prior configuration.

Given these differences in access to flight data, there was an essential need to create a universal measuring system that could be integrated into various types of UAV. Such a system would enable data acquisition independently of the UAV brand, its level of autonomy, type of control unit, control station, or software firmware. DroneBox, protected by patent, represents a unique measuring system developed as a network of individual hardware modules. Its design was optimized for obtaining time-synchronized data necessary for monitoring and evaluating the flight characteristics of UAV's. The development of DroneBox prototypes underwent several iterations, with the final version successfully applied and validated on a 3DR IRIS+ type UAV. In addition to examining the impact of various factors, such as increased weight, the effect of gusty wind on propulsion units, or the function of the autopilot system, DroneBox also enables the creation of operational envelopes for UAV's, providing a detailed view of their flight characteristics and behavior in real conditions. By extending DroneBox with a Long Term Evolution (LTE) module and GPS, it could become an alternative flight transponder for UAS and provide a redundant telemetry device for the flight operator. This would allow for more effective tracking of the UAV's position, enhancing overall safety and efficiency of flight operations.

The created operational envelope for the IRIS+ quadcopter defines the area where this UAV can operate without losing altitude. The envelope was determined unilaterally, considering only negative pitch. However, assuming the uniformity of the geometry of the quadcopter, or symmetric UAV's, it could also be valid symmetrically for positive angles of attack. Such a solution, which allows for quick orientation in basic flight states and fundamental parameters like thrust, angle of atack, and speed, is not commonly used according to current literature, though it has significant potential. Implementing an operational envelope can provide UAV operators with important information for quick orientation and understanding of the dynamic state in which the UAV is located.

Incorporating points of energy efficiency into the operational envelope could contribute to determining the boundary points of UAV operational capabilities or the maximum speed of the UAV. Such an approach could be significantly enhanced with data from wind tunnel tests, which would

allow for the optimization of the operational envelope. Since the data were obtained at low flight altitudes, it is likely that the operational envelope would exhibit different characteristics at other altitudes. Future research, therefore, could expand the envelope to include another dimension – flight altitude, leading to more comprehensive and accurate models of operational envelopes.

The DroneBox system, used for collecting flight data, proved to be an effective tool for creating operational envelopes and dynamically adapting them based on current operational conditions and UAV performance. Its ability to record and analyze detailed flight data opens up new possibilities for the development of advanced systems for monitoring UAV flight operations.

V. CONCLUSION

Research focused on creating and validating a system designed for monitoring and evaluating UAV's flight and performance characteristics represents a significant contribution to testing and analyzing quadcopters. The aim of the study was to develop and verify new methods and tools that would allow for the optimal utilization of the flight properties of multicopters.

Within the context of real-environment flight tests, the functionality of the proposed measuring system, known as DroneBox, was successfully validated. This system demonstrated its ability to accurately record and analyze the flight parameters of the quadcopter, thereby confirming its reliability and significance for future applications. The results obtained from DroneBox showed a high correlation with data from the Pixhawk control unit. An indisputable advantage of the system is the application of a Pitot-static tube with modified inlet geometry. This allows for the compensation of angle of attack error, making it suitable for measuring indicated airspeed, which is necessary for the correct assessment of the performance parameters of the quadcopter in real flight conditions. On the other hand, the device's limitations include a relatively low sampling frequency and high sensitivity to electromagnetic induction, which occurs near the motors of the UAV and also near power conductors. Electromagnetic induction caused errors or failures in measuring the magnetometer and interruptions in Hall sensors.

A series of simulated flights in the robotic simulation program Gazebo provided the basis for creating an operational envelope for the quadcopter. This envelope, representing the relationship between the angle of attack, flight speed, and revolutions of the propulsion units, is a key contribution of this work. The operational envelope was created as a tool for predicting the performance parameters of the quadcopter necessary for flight on the horizon at specific weights of the quadcopter. The creation of the operational envelope was based on simulated data and then compared with data from real flights recorded on the Pixhawk control unit. This integrated approach offers new possibilities for understanding and optimizing the flight properties of these devices.

REFERENCES

[1] *ICAO Unmanned Aircraft Systems (UAS), Circular 328*, Int. Civil Aviation Org., Montreal, QC, Canada, 2011.
[2] *Code of Federal Regulations, Title 14, Part 107*, Federal Aviation Admin., Washington, DC, USA, 2016.
[3] *Global UAV Market 2022–2026 Report*, Market Rep. World, London, U.K., Aug. 2022.
[4] L. Shi, N. J. H. Marcano, and R. H. Jacobsen, "A review on communication protocols for autonomous unmanned aerial vehicles for inspection application," *Microprocessors Microsystems*, vol. 86, Oct. 2021, Art. no. 104340.
[5] D. Jung and P. Tsiotras, "Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle," in *Proc. AIAA Infotech@Aerospace Conf. Exhib.*, May 2007, p. 2768.
[6] A. I. Hentati, L. Krichen, M. Fourati, and L. C. Fourati, "Simulation tools, environments and frameworks for UAV systems performance analysis," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 1495–1500.
[7] K. Christodoulou, M. Vozinidis, A. Karanatsios, E. Karipidis, F. Katsanevakis, and Z. Vlahostergios, "Aerodynamic analysis of a quadcopter drone propeller with the use of computational fluid dynamics," *CET J.-Chem. Eng. Trans.*, vol. 76, pp. 181–186, Oct. 2019.
[8] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. Von Stryk, "Comprehensive simulation of quadrotor uavs using ros and gazebo," in *Proc. 3rd Int. Conf. Simulation, Modelling, Program. Auto. Robots (SIMPAR)*, Tsukuba, Japan. Cham, Switzerland: Springer, Nov. 2012, pp. 400–411.
[9] C. R. Russell, J. Jung, G. Willink, and B. Glasner, "Wind tunnel and hover performance test results for multicopter UAS vehicles," in *Proc. Amer. Helicopter Soc. (AHS) Int. Annu. Forum Technol. Display*, 2016, Paper no. ARC-E-DAA-TN31096.
[10] E. Baris, C. P. Britcher, and G. Altamirano, "Wind tunnel testing of static and dynamic aerodynamic characteristics of a quadcopter," in *Proc. AIAA Aviation Forum*, Jun. 2019, p. 2973.
[11] V. Praveen and S. Pillai, "Modeling and simulation of quadcopter using PID controller," *Int. J. Control Theory Appl.*, vol. 9, no. 15, pp. 7151–7158, 2016.
[12] M. Schmittle, A. Lukina, L. Vacek, J. Das, C. P. Buskirk, S. Rees, J. Sztipanovits, R. Grosu, and V. Kumar, "OpenUAV: A UAV testbed for the CPS and robotics community," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Physical Syst. (ICCPS)*, Apr. 2018, pp. 130–139.
[13] A. Shahoud, D. Shashev, and S. Shidlovskiy, "Visual navigation and path tracking using street geometry information for image alignment and servoing," *Drones*, vol. 6, no. 5, p. 107, Apr. 2022.
[14] B. Theys, G. Dimitriadis, T. Andrianne, P. Hendrick, and J. De Schutter, "Wind tunnel testing of a VTOL MAV propeller in tilted operating mode," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, May 2014, pp. 1064–1072.
[15] S. Bhandari, P. Navarro, and A. Ruiz, "Flight testing, data collection, and system identification of a multicopter UAV," in *Proc. AIAA Model. Simulation Technol. Conf.*, Jan. 2017, p. 1558.
[16] B. Theys and J. De Schutter, "Forward flight tests of a quadcopter unmanned aerial vehicle with various spherical body diameters," *Int. J. Micro Air Vehicles*, vol. 12, Jan. 2020, Art. no. 175682932092356.
[17] S. Kušmírek, V. Socha, L. Hanáková, K. Hylmar, R. Matyáš, and P. Kubový, "Validation of the manual indoor UAV flying characteristics methodology," in *Proc. New Trends Civil Aviation (NTCA)*, Nov. 2020, pp. 129–133.
[18] *PX4 User Guide: Pixhawk Series Manual*, PX4 Develop. Team Community, Zürich, Switzerland, 2023.
[19] S. Kusmirek, V. Socha, D. Urban, L. Hanakova, F. Kripsky, R. Walton, and P. Pecho, "Design of pitot-static tube shapes and their influence on airspeed measurement accuracy," *J. Aerosp. Eng.*, vol. 37, no. 4, Jul. 2024, Art. no. 04024045.
[20] D. Serrano, "Introduction to ROS-robot operating system," Open Source Robot. Found., Mountain View, CA, USA, Tech. Rep. OSRF-TR-2011-01, 2011.
[21] A. Koubâa, *Robot Operating System (ROS)*, vol. 1. Springer, 2017.
[22] K. Dang Nguyen and T.-T. Nguyen, "Vision-based software-in-the-loop-simulation for unmanned aerial vehicles using gazebo and PX4 open source," in *Proc. Int. Conf. Syst. Sci. Eng. (ICSSE)*, Jul. 2019, pp. 429–432.
[23] F. Furrer, M. Burri, M. Kamel, J. Nikolic, and M. Achtelik. (2015). *PX4 SITL Gazebo Classic: Gazebo Motor Model Source Code*. [Online]. Available: Px4-sitl_gazebo-classic/src/gazebo_motor_model.cpp

[24] M. Bangura, M. Melega, R. Naldi, and R. Mahony, "Aerodynamics of rotor blades for quadrotors," 2016, *arXiv:1601.00733*.
[25] P. Martin and E. Salaun, "The true role of accelerometer feedback in quadrotor control," Decis. Control Lab., Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep. DCL-2010-03, 2010.
[26] R. W. Deters, S. Kleinke, and M. S. Selig, "Static testing of propulsion elements for small multirotor unmanned aerial vehicles," in *Proc. 35th AIAA Appl. Aerodynamics Conf.*, Jun. 2017, p. 3743.
[27] M. Awadalla, Y. Y. N. Yannick, and K. A. Asbeh, "Modeling the dependence of barometric pressure with altitude using caputo and Caputo–Fabrizio fractional derivatives," *J. Math.*, vol. 2020, pp. 1–9, Nov. 2020.
[28] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *J. Amer. Stat. Assoc.*, vol. 107, no. 500, pp. 1590–1598, Dec. 2012.
[29] M. Lavielle, "Using penalized contrasts for the change-point problem," *Signal Process.*, vol. 85, no. 8, pp. 1501–1510, Aug. 2005.

STANISLAV KUSMIREK received the B.S. degree in prosthetics and orthotics from the Technical University of Kosice, Slovakia, in 2013, and the M.S. degree in biomedical and clinical technology and the Ph.D. degree in air traffic control and management from the Czech Technical University (CTU) in Prague, Czech Republic, in 2015 and 2024, respectively.

In 2016, he was a Researcher with the Institute of Photonics and Electronics, Czech Academy of Sciences. From 2017 to 2019, he was a Mechanical Engineer and a Designer for Mercedes Benz Technologies. Since 2018, he has been a Ph.D. Candidate in air traffic control and management with the Faculty of Transportation Sciences, Czech Technical University in Prague, where he is also a Researcher with the Laboratory of Human Factors and Automation in Aviation. In 2019, he completed a six-month internship with the Vertical Lift Research Center of Excellence Penn State University, USA. His main research interests include testing of unmanned aerial vehicles, flight mechanics, and computational flow dynamics.

VLADIMIR SOCHA (Senior Member, IEEE) received the degree in air transport management and prosthetics and orthotics from the Technical University of Kosice, Slovakia, in 2011, the M.S. degree in biomedical and clinical technology from the Faculty of Biomedical Engineering, Czech Technical University (CTU) in Prague, Czech Republic, in 2013, and the Ph.D. degree in transportation from the Faculty of Aeronautics, Technical University of Kosice, in 2016.

Since 2013, he has been a Research Assistant with the Faculty of Biomedical Engineering, CTU in Prague. In 2016, he expanded his responsibilities by joining the Department of Air Transport, CTU in Prague, where he established and currently oversees the operations of the Laboratory of Human Factors and Automation in Aviation. He was appointed as an Associate Professor with CTU in Prague, in 2018. Presently, he holds the position of the Head of the Laboratory, the Deputy Head of the Department of Air Transport, CTU in Prague, and the Deputy Vice-Dean for Science and Research of the Faculty of Transportation Sciences, CTU in Prague. He has authored more than a hundred academic publications, mentored numerous doctoral students, organized international scientific conferences, and collaborated with numerous national and international institutions in the field of aviation. His areas of research interests include signal processing, automation in aviation, including unmanned aviation, aviation physiology, human–machine interface, biomedical and clinical technologies, and statistics and machine learning.

TOMAS MALICH received the B.S. degree in software technologies and management from the Faculty of Electrical Engineering, Czech Technical University in Prague, and the Engineering degree in air traffic management and control from the Faculty of Transportation Sciences, Czech Technical University in Prague. From 2015 to 2024, he was a Developer and the Manager for company GK Software AG. He mainly focused on the development of iOS applications, Java backend applications, and AI technologies, such as machine learning, data analysis, and management. His research is also mainly focused on those areas but for practical use in air traffic.

LUBOS SOCHA received the M.S. degree in supersonic aircraft piloting from the High Military School, Košice, in 1985, and the Ph.D. degree in operational and combat utilization of aviation from the Military Aviation Academy, Košice, in 2003.

From 1985 to 1998, he was a Pilot in the military air force, holding various roles. From 1998 to 2004, he was an Assistant Professor and the Head of the Air Force Operational Use Department, Military Aviation Academy. Following the transformation of the Military Aviation Academy into a Civilian Faculty of Aviation within the framework of the Technical University of Kosice, he held the position of the Department Head for Air Traffic Management, until 2005. From 2006 to 2007, he assumed the role of the Director of the Office for Defense Standardization, Codification, and State Quality Assurance, Trenčín. Since 2008, he has been actively contributing to the Aviation Faculty with the Technical University of Kosice. He attained the position of an Associate Professor with the Technical University of Kosice, in 2019. Since 2021, he has also been part of the Department of Air Transport, Faculty of Transportation Sciences, Czech Technical University in Prague. He has authored more than 220 publications, including three monographs and several university textbooks.

KAREL HYLMAR received the B.S. and M.S. degrees in air transport and air traffic control and management from Czech Technical University (CTU) in Prague, in 2020 and 2022, respectively, where he is currently pursuing the Ph.D. degree in air traffic control and management with a specialization in unmanned aerial vehicles. Since 2022, he has been an Integral Member with the Laboratory of Human Factors and Automation in Aviation, CTU in Prague, where he is a Researcher.

LENKA HANAKOVA received the B.S. and M.S. degrees in biomedical and clinical technology and the Ph.D. degree in air traffic control and management from Czech Technical University (CTU) in Prague, Czech Republic, in 2014, 2016, and 2023, respectively.

Since 2017, she has been a member with the Laboratory of Human Factors and Automation in Aviation, CTU in Prague, where she is currently a Researcher. Additionally, she is the Head of the Human Factors Research Team, Department of Air Transport, CTU in Prague. Her research interests include signal processing, human–machine interface, automation in aviation, and aviation physiology.

Dr. Hanakova was awarded the Best Student Paper from the 38th International Conference on Telecommunications and Signal Processing, in 2015. In 2018, she received the Josef Hlavka Award for the best students of Prague public universities.