

PADRE – Propeller Anomaly Data REpository for UAVs various rotor fault configurations

Radosław Puchalski¹, Marek Kołodziejczak¹, Adam Bondyra¹, Jinjun Rao², Wojciech Giernacki¹

Abstract—The article presents a drone sensory database collected during flights with different types of propeller failures. Measurements from four accelerometers and four gyroscopes were collected during 20 flights with two types of faults occurring in different configurations in one, two, three or four rotors. The paper shows the architecture of the system and the procedure for acquiring and processing the data. Raw sensor outputs, pretreated data, and digitally processed signals were provided in a publicly available repository, the structure and purpose of which are discussed in the paper. The applicability and potential use of the shared data for other research are indicated. The provided repository should be helpful in developing methods for detecting and classifying faults in actuators of unmanned aerial vehicles (UAVs). It will be particularly useful for researchers working on data-driven methods. The default purpose of the dataset is to train artificial intelligence models that require large amounts of data.

SUPPLEMENTARY MATERIAL

https://github.com/AeroLabPUT/UAV_measurement_data
<https://uav.put.poznan.pl/archives/763>

I. INTRODUCTION

Unmanned aerial vehicles are becoming more widely used. They are adopted in agriculture [1], [2], an inspection of building structures [3], [4] or power installations [5], [6], transportation [7], [8], [9], the military [10], [11] or for ordinary recreation among other applications [12], [13]. They increasingly often operate in close proximity to humans [14]. This requires ensuring the highest safety standards for their use. Therefore, it is very important to quickly detect emerging irregularities in the operation of drone systems. Recent scientific advances have placed great emphasis on effective methods of detecting defects in the components of flying vehicles [15].

Today's machine learning techniques typically require extensive data to train a model [16]. In order to perform well on future tasks, the model needs to process as many similar, yet not identical, examples as possible. Often, collecting and pretreating data is the most time-consuming process [17]. The training process itself is largely automated, while it takes a lot of work to accumulate, purify, aggregate, normalize, and

¹R. Puchalski, M. Kołodziejczak, A. Bondyra, W. Giernacki are with the Institute of Robotics and Machine Intelligence, Faculty of Control, Robotics and Electrical Engineering of Poznań University of Technology, Poznań, Poland, corresponding author: radoslaw.puchalski@doctorate.put.poznan.pl, <http://www.uav.put.poznan.pl>.

²J. Rao is with the Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China.

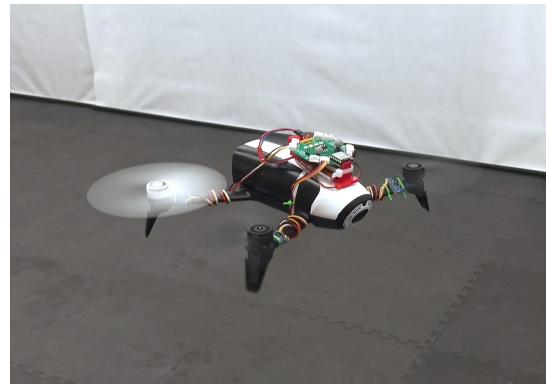


Fig. 1. *Bebop 2* drone equipped with additional measurement electronics during the data collection experiment.

standardize the dataset. Especially in the work of a researcher and scientist, who should focus on scientific methods, the process of data preparation is not a valuable stage, but it cannot be skipped.

The aim and main contribution of the present work is to prepare a repository with in-flight sensory data of unmanned aerial vehicles. The database is designed primarily to train and test artificial intelligence models that detect and classify damage to UAV actuators. In the considered example, damages of the propeller blades are concerned. It was decided to develop a universal and airframe-independent system that would record sensor readings during flight. An exemplary flight in which data acquisition was carried out is shown in Fig. 1.

The article is organized as follows: Section II discusses an overview of related work and the state of the art in the field. The design and software concept of the data acquisition system are described in Section III. Additionally, information on digital signal processing can be found there. The structure of the repository, with a precise description of its contents, is presented in detail in Section IV. A summary of the work and suggestions for future plans are given in Section V.

II. RELATED WORK

The unmanned aerial vehicle is a convenient flying platform for collecting data of various types. The following are some examples of research on creating data repositories. These are usually data acquired from the onboard camera, sometimes from sensors with which the vehicle is equipped by default. However, there is a lack of databases that can be used in studies regarding rotor fault detection.

One can encounter works that aim to create a UAV data repository. Antonini *et al.* in [18] present a large-scale dataset of inertial sensors and ground data from a platform equipped with an inertial measurement unit (IMU) and rotor tachometers, as well as virtual cameras. The authors indicate the usefulness of the prepared data for developing algorithms for visual inertial navigation or 3D reconstruction.

Mylonas *et al.* in [19] built an agricultural dataset that includes proximal and aerial images of various crops, weeds, and disorders such as diseases and pests that occur in agriculture. The collected data are expected to be useful in various field applications based on vision and artificial intelligence.

In [20], Johnson *et al.* used UAVs to take orthorectified images. Such images may find use in the map development. They have developed an open repository and are urging other drone users to expand it with their materials.

In recent years, many research centers have taken up the field of detecting anomalies in the operation of flying robot components. Typically, methods for detecting such abnormalities are divided into those based on a mathematical model of the object and model-free ones [21]. Particularly in the case of the latter, called data-based or data-driven, historical data are required to prepare the model [22], [23]. On this basis, an algorithm is prepared analyzing current measurement data, finds similarities to the samples it has been trained on. The more such data has been provided to the trained model, the better it can be prepared to recognize future malfunctions.

Vibrational data are often used for early warning systems detecting drone anomalies. An example of such a study is the work of Ghazali *et al.* [24]. The authors used data from various accelerometers to detect arm damage, propeller imbalance and partial propeller loss.

Al-Haddad *et al.* in [25] took inertial data from the accelerometers by wire while flying the drone and then used a discrete wavelet transform and a deep neural network to classify the damage to the propellers.

A similar idea of designing a repository for UAV fault detection and isolation was presented by Keipour *et al.* in [26]. The authors collected data in various formats from flights of Carbon-Z T-28 fixed-wing aircraft. The data cover eight different faults and were collected in a total of 66 minutes of flight time. The representation of individual faults and anomalies is not evenly distributed. The repository also contains a lot of data not processed in any way. The main difference of the database presented in our article is the type of UAV used for data collection, multirotors instead of fixed-wing aircraft. The advantage of our solution is the equal time and identical flight conditions for each scenario. All data were collected in the same form and are convenient for direct use to train the artificial intelligence model.

III. STAND-ALONE DATA ACQUISITION SYSTEM

Various commercial drones and designs developed in the lab have different sets of onboard sensors. Some of these readings are easily accessible, while others cannot be used

TABLE I
MICROCONTROLLER PARAMETERS [27].

MCU	STM32H743IIT6
Core architecture	ARM Cortex-M7
Core frequency	480 MHz
Flash memory	2 MiB
RAM memory	1 MiB
Package	LQFP176
Supply voltage	1.62–3.6 V
Communication interfaces	4 × I ² C, 4 × UART, 6 × SPI, 4 × SAI, 2 × SD/SDIO/MMC, 2 × CAN, 2 × USB OTG, SPDIFRX, MDIO, one-wire SWPPI I/F, Ethernet, HDMI-CEC, camera

in custom applications. As standard, they are equipped with a central unit that measures the behavior of the flying robot only from the UAV's fuselage. Detection of faults in the drone's rotors will be more effective the closer the sensor is placed to the faulty part.

A. Hardware

An accelerometer and gyroscope data acquisition system has been prepared for the task of building a UAV measurement data repository. It is based on an STM32 microcontroller from the H7 family. The basic parameters of the applied chip are presented in Table I.

On the PCB board, the terminals have been placed to connect 4 inertial modules. Each module was routed to a different I²C bus, allowing simultaneous acquisition without waiting for the previous reading to finish. It has been optimized for cooperation on a quadrotor. Of course, acquisition from a smaller number of IMUs is possible. Using different addresses of measurement modules connected to a single I²C interface, it is also possible to carry out acquisition from a larger number of sensors. Data are written to a micro SD card connected via a 4-bit SDIO interface. A 6-pin SWD connector is used to program the chip. A simple interface with the user is provided by a 4-channel microswitch and four LEDs. Leads for WiFi and Bluetooth modules were also designed, but their use proved unnecessary at this stage of the work. Their application is planned for future research. Connectors for digital microphones have also been prepared, whose use is planned for expansion of the existing repository. The chip is powered by a voltage converter from a 3.7 V Li-Po battery with a capacity of 250 mAh. The current consumption of all circuits does not exceed 200 mA. The layout of the top layer is shown in Fig. 2. The PCB was attached to the drone using a bracket made in a 3D printer. Considering the layout size (ca. 66 x 66 mm), it can be applied to most types of drone structures, excluding the smallest models, on which it will not fit. In particular, it is designed for multirotors.

MPU-6500 modules equipped with a DMP coprocessor are used for linear acceleration and angular velocity measurements. They were attached to the arms in close proximity to the rotors. In the *Parrot Bebop 2* UAV, the rear modules were fixed parallel to the ground, while the front ones were tilted

TABLE II
IMU SENSOR PARAMETERS [28].

IMU	MPU-6500
ADC Word Length	16 bit
Communication interfaces	I ² C, SPI
Supply voltage	1.17–3.45 V
Accelerometer	
Number of axes	3
Range	2, 4, 8, 16 g
Sensitivity	2048...16384 LSB/g
Output data rate	4000 Hz
Gyroscope	
Number of axes	3
Range	250, 500, 1000, 2000 dps
Sensitivity	16.4...131 LSB/dps
Output data rate	8000 Hz

Fig. 2. The top layer of the developed PCB used to collect data from UAV flights with damaged propellers.

toward the front of the drone by about 19 degrees. This was forced by the shape of the front arms, the upper surfaces of which are aligned at such angles. The most important parameters of the modules used are presented in Table II.

B. Software

The STM32CubeIDE development environment was used to prepare the software controlling the operation of the microcontroller. It allows both to configure all the peripherals, as well as to write, compile, debug and upload the code. The main program was written in C using the HAL library. The core functions of the program are sensor configuration (setting ranges and acquisition rates), reading data from sensor registers, and writing to the corresponding file on the memory card. A simplified algorithm for program operation is shown in Fig. 3.

C. Digital signal processing

Data from sensor registers are saved to the memory card in raw hexadecimal numbers. To obtain useful values, they are converted to the signed integer type (*int16*) and then normalized to the floating-point type with values ranging from -1 to $+1$. This form can be used directly in the process of training an artificial intelligence model, such as an artificial neural network. In addition, to visualize the actual values of linear accelerations and angular velocities, the data were converted to values scaled in units of ground acceleration [g] (accelerometer) and degrees per second [dps] (gyroscope). Data were also prepared in the form of spectra obtained with a fast Fourier transform. Therefore, it is possible to directly assess how the different fault classes are represented in the frequency domain. All data collected refer to the 3 accelerometer and 3 gyroscope axes of each of the 4 sensor modules mounted on the drone's arms (Fig. 4).

IV. DATA REPOSITORY

The measurement data were collected and made available as a repository on GitHub. A project named

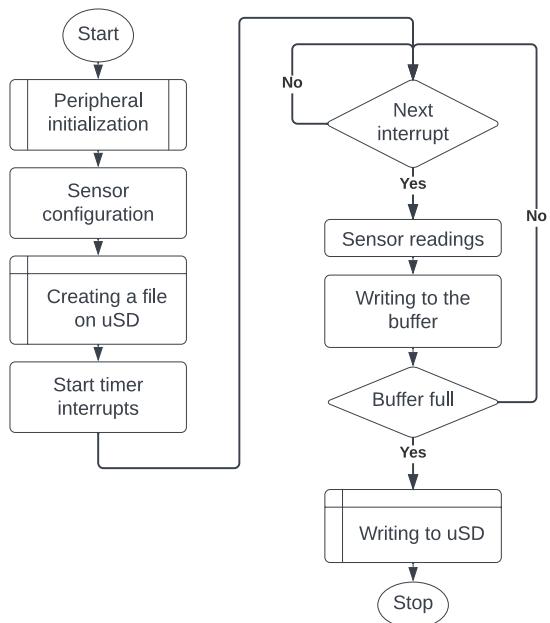


Fig. 3. The program operation algorithm

UAV_measurement_data was created, with the files categorized into a hierarchical structure. Only measurements from the sensors with which the independent data acquisition system was equipped were made available. No control signals or other data were recorded. However, nothing prevents the repository from being expanded in the future to include, for example, ground truth data or EO/IR data. The security of the database, its resilience and safety were ensured by storing copies of the data in two other repositories and on a local disk.

At the time of writing, the repository contains a folder with measurements from the *Parrot Bebop 2* drone, with subfolders containing sensory data in various forms. The basic parameters of the flying vehicle used the acquisition process are shown in Table III.

In the file naming convention, information about the class of damage in the form of four digits was included. These digits represent the four propellers in turn: A, B, C, D. The digit 0 indicates a working propeller, 1 – a propeller with

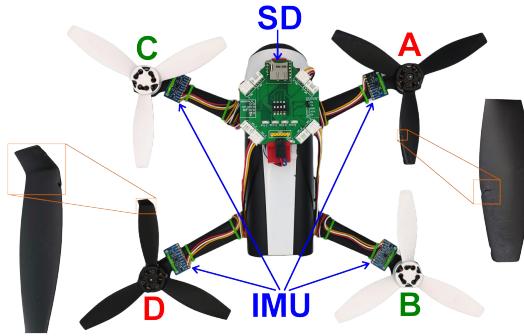


Fig. 4. *Parrot Bebop 2* with a data acquisition system. Close-ups show two types of propeller damage: a chipped edge and a bent tip.

TABLE III
UAV PARAMETERS [29].

Drone	Parrot Bebop 2
Weight	500 g
Dimensions	328 x 328 x 89 mm
Propeller size	15.2 cm
Battery capacity	2700 mAh
Flight time	up to 25 minutes
Maximum horizontal speed	16 m/s
Maximum ascent speed	6 m/s
WiFi range	300 m

a chipped edge, and 2 – a propeller with a bent tip. For example, class 1120 means that propellers A and B had a chipped edge, propeller C had a bent tip, and propeller D was undamaged. The location of the data acquisition system along with the inertial sensors in the drone and the appearance of the damaged propellers are shown in Fig. 4.

Individual flights were not identical and the drone was operated manually using a wireless controller. Various maneuvers were performed, but most flights included hovering, sideways flight, ascending and descending, diagonal paths, circle, and eight-shaped trajectories.

Data from 20 fault classes were prepared:

- all propellers operable: 0000,
- single propeller damaged: 1000, 0100, 0010, 0001, 2000, 0200, 0020, 0002,
- two propellers damaged: 1100, 1020, 1002, 0120, 0102, 0022,
- three propellers damaged: 1120, 1102, 1022, 0122,
- all propellers damaged: 1122.

A. Raw data

Raw_data directory contains 20 CSV files, each matching to a flight with a different fault class. The name of each file contains information about the drone with which the flight was made, the measurement ranges of the accelerometers and gyroscopes, and the class name in four digits. Each file contains 86016 lines corresponding to 172.032 seconds of flight for a sampling rate of 500 measurements per second. Every line in turn contains 96 ASCII characters, which are the values of the measurement registers stored in hexadecimal format. Each data byte occupies 2 HEX

symbols. Individual axis consists of 2 bytes of data: H – most significant byte, L – least significant byte. The values on each line are arranged in order:

```
A_ax_H, A_ax_L, A_ay_H, A_ay_L, A_az_H, A_az_L,
A_gx_H, A_gx_L, A_gy_H, A_gy_L, A_gz_H, A_gz_L,
B_ax_H, B_ax_L, B_ay_H, B_ay_L, B_az_H, B_az_L,
B_gx_H, B_gx_L, B_gy_H, B_gy_L, B_gz_H, B_gz_L,
C_ax_H, C_ax_L, C_ay_H, C_ay_L, C_az_H, C_az_L,
C_gx_H, C_gx_L, C_gy_H, C_gy_L, C_gz_H, C_gz_L,
D_ax_H, D_ax_L, D_ay_H, D_ay_L, D_az_H, D_az_L,
D_gx_H, D_gx_L, D_gy_H, D_gy_L, D_gz_H, D_gz_L.
```

The arrangement of the axis of each sensor relative to the individual rotors is shown in Fig. 5.

B. Range data

A folder named Range_data is organized in a similar way. It contains data from accelerometers converted to standard gravity [g] and from gyroscopes expressed in degrees per second [dps]. These data are primarily suitable for manual analysis, as it allows a clear comparison of sensor readings for different classes of actuator damage. The files in this subfolder also contain 86016 lines, each of which includes floating-point data in the following manner:

```
A_ax, A_ay, A_az, A_gx, A_gy, A_gz,
B_ax, B_ay, B_az, B_gx, B_gy, B_gz,
C_ax, C_ay, C_az, C_gx, C_gy, C_gz,
D_ax, D_ay, D_az, D_gx, D_gy, D_gz.
```

Differences in the readings of selected sensors in values corresponding to the set measurement ranges of the accelerometer and gyroscope during flight with efficient propellers compared to the fault occurring are shown in Fig. 6 and Fig. 7.

C. Normalized data

The one most suitable for training artificial intelligence models are data stored in a directory called Normalized_data. It contains the same data as the Range_data folder but is normalized to a range of -1 to +1. For accelerometer data, a value of -1 corresponds to an acceleration of -16 g and +1 is +16 g. Similarly, for the gyroscope measurements, the values range from -1000 dps to +1000 dps, respectively. Such data can be used directly, for example, in the process of training an artificial neural network. These are the nominal values typically used in this machine learning technique [30]. It would be an equally good idea to use the range from 0 to +1, but due to the direct indication of the linear acceleration or angular velocity direction, it was decided to use signed values.

D. FFT data

In addition to pure sensor readings, both in raw and pre-processed form, it was also decided to expand the repository with data subjected to digital signal processing operations. As a first step, the data were converted from the time form to the frequency domain via a fast Fourier transformation (FFT_data directory). A window function was also applied to reduce the spectral leakage effect [31]. The data were placed in a folder whose name represents the following parameters:

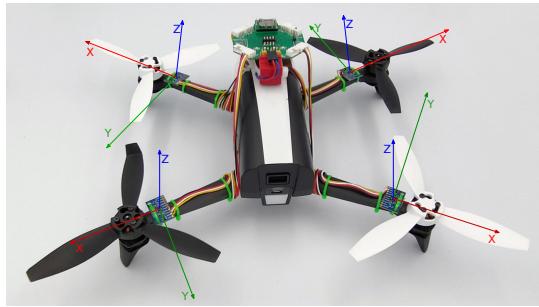


Fig. 5. IMU local coordinate systems.

- frame length – number of samples subjected to Fourier transformation,
- the window function used,
- the lower frequency range,
- the upper frequency range.

The last parameter determines the frequency cut-off point. It was decided to discard the lowest and highest frequencies of the analyzed signal, which significantly reduces file size and model training time. For example, for a sampling frequency of 500 Hz and a measurement window length of 128 samples, 64 real bars corresponding to successive frequencies from 0 to 250 Hz are obtained with an interval of about 3.9 Hz. If frequencies below 74.2Hz (bar 20) and above 203.1 Hz (bar 52) are discarded, 33 most useful frequencies remain.

It has been proven that the fault characteristic frequency occurs in a specific band. Previous studies using data in this form show the correctness and appropriateness of their use [32]. The selection of sample ranges was based on analysis of the spectrograms (see Fig. 8 and Fig. 9). The files contain data from one type of a sensor. In this case, it was decided to separate the accelerometer data from the gyroscope data. Each line of the file contains in turn the values of each bar for sensor A, then for sensor B, C and D. When using 4 sensors with 3 axes each and 33 useful frequencies, this gives 396 values per line. The number of lines for raw data with 86016 readings and a measurement window length of 128 samples is $86016/(128 \cdot 2) + 1 = 337$.

The desirability of using the data in the frequency domain is shown in Fig. 10 and Fig. 11. They clearly present the differences in signal frequencies between the different classes of faults. Noticeable 12 characteristic frequencies correspond to the X, Y, and Z axes of sensors A, B, C, and D, respectively.

E. Further data

Over time, the repository will be successively supplemented with data obtained during another experiments, but its basic layout has been established. The use of different Fourier transform parameters, other methods to extract useful signal features, as well as measurements taken from other flying robot designs are planned. Each folder includes a *readme.txt* file that explains the directory layout and file contents. The organization of the repository in graphical form is shown in Fig. 12.

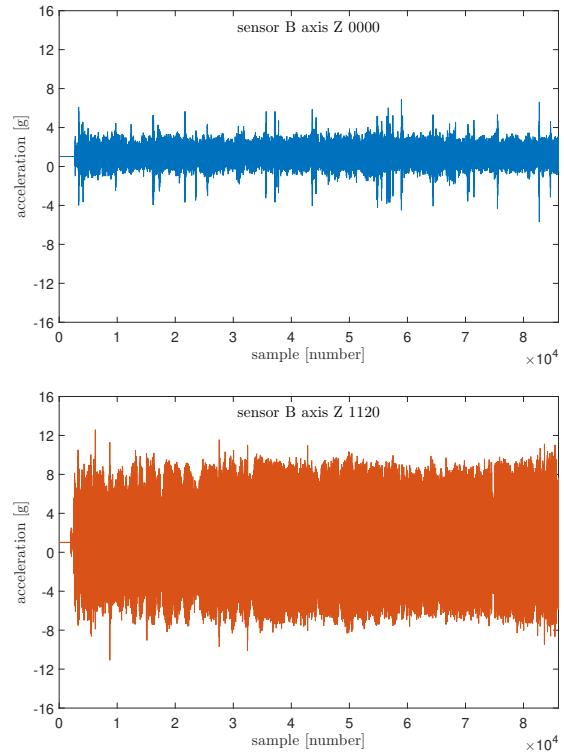


Fig. 6. Z-axis readings from the A accelerometer during a flight with all working propellers (0000) and three faulty ones (1120).

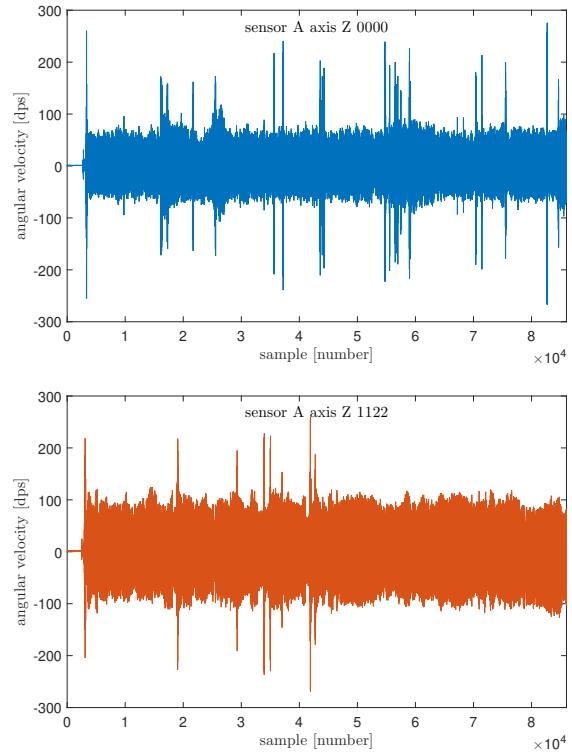


Fig. 7. Z-axis readings from the B gyroscope during a flight with all working propellers (0000) and four faulty ones (1122).

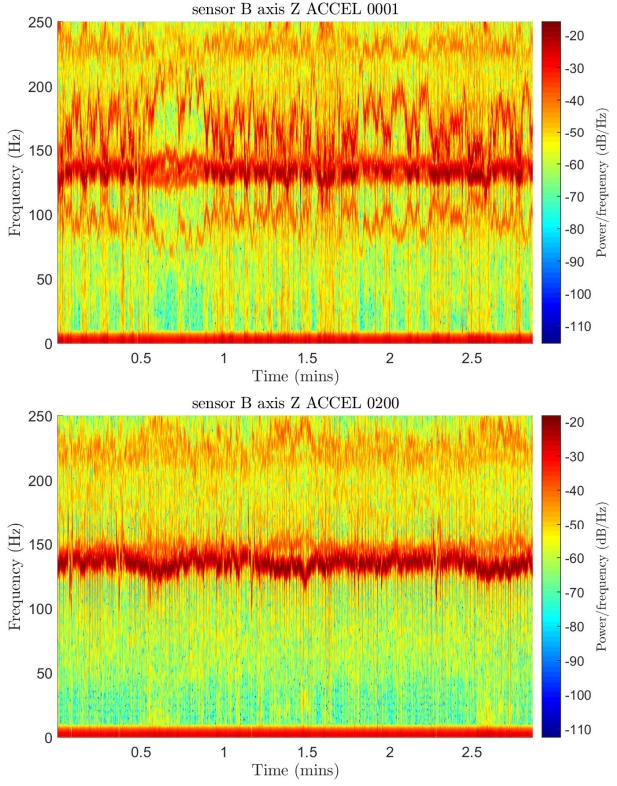


Fig. 8. Spectrograms of the Z-axis readings of the B unit accelerometer during 0001 and 0200 scenario flights.

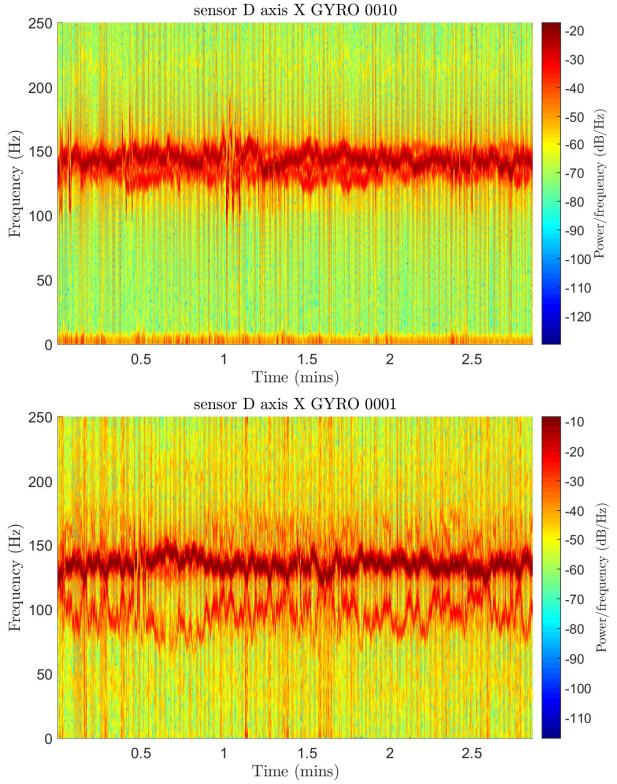


Fig. 9. Spectrograms of the X-axis readings from the D unit gyroscope during a 0010 and 0001 scenario flights.

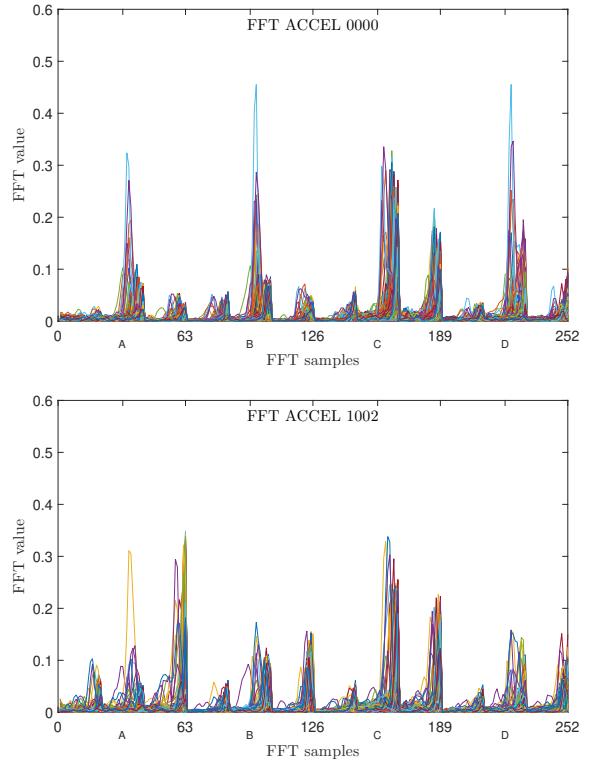


Fig. 10. FFT data of the four accelerometers during a 0000 and 1002 scenario flights.

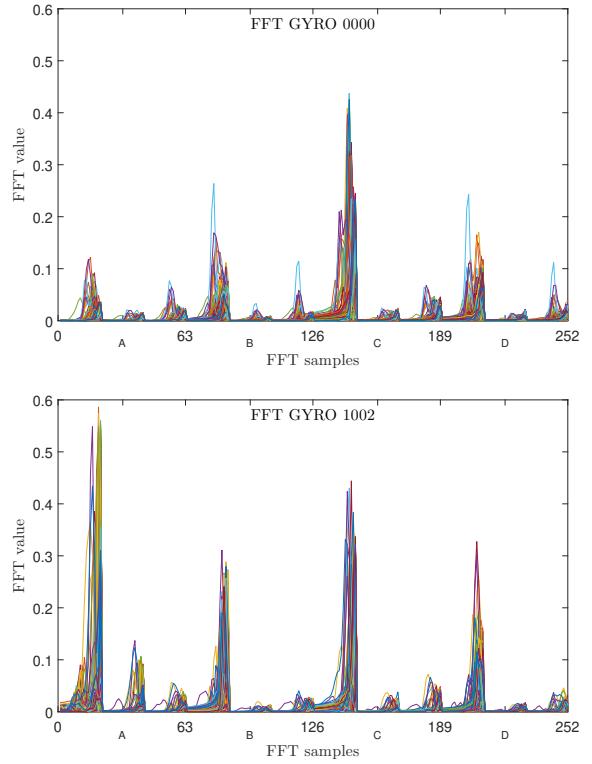


Fig. 11. FFT data of the four gyroscopes during a 0000 and 1002 scenario flights.

Ultimately, the same system will be used to detect and classify multirotor propeller faults. Data from the repository will be used to train the model, which will be implemented in the computationally efficient microcontroller used. All calculations will take place in real-time.

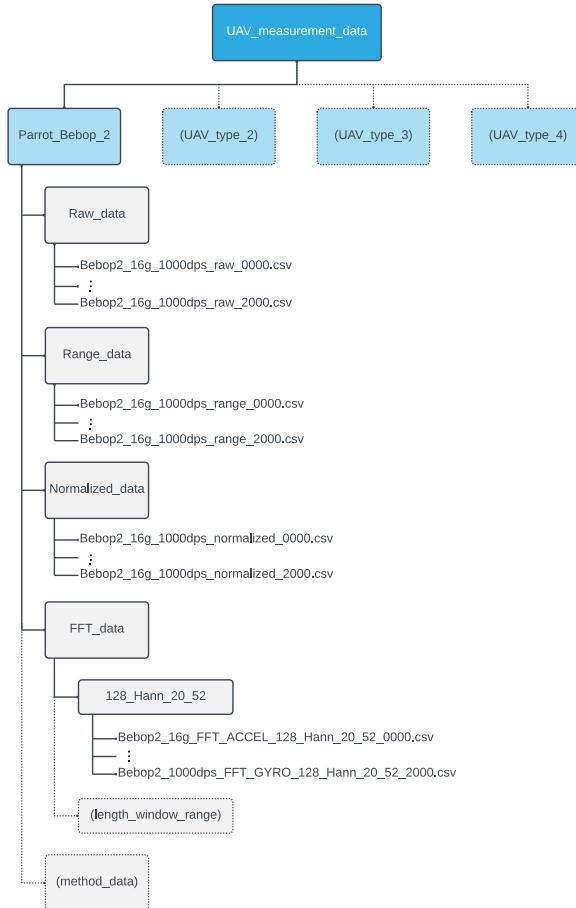


Fig. 12. Repository structure.

V. CONCLUSION

During the development of the repository, measurement data were collected from four accelerometers and gyroscopes located near each of the quadrotor's four propellers. Twenty experimental flights were conducted in various propeller fault configurations. Acquisitions were made on 1 flight with operable propellers, 8 flights with one damaged propeller (two types of damage, four fault locations), 6 flights with two faults, 4 flights with triple faults and 1 flight with four damaged propellers.

The datasets included in the proposed repository can be useful for various purposes. The authors are concerned with the topic of UAV damage detection and classification, and the repository was created with this objective in mind. However, it is not difficult to imagine the usefulness of the collected measurements in other applications. For example, by using frequency data, it is possible to design the optimal shape and size of rotors, generating the least vibrations. Using the data represented in units of linear acceleration, one can

determine the overloads to which the arms of the multirotor are exposed. But above all, the data will be useful for creating artificial intelligence models for detecting rotor damage and determining the type and location of its occurrence.

Future work will include enhancing the repository with data taken from other flying robots. It is also planned to undertake acquisition from other types of sensors and use the developed system to perform real-time fault detection and classification.

ACKNOWLEDGMENT

This research was financially supported as a statutory work of Poznan University of Technology (grant no. 0214/S-BAD/0241).

REFERENCES

- [1] D. Rakesh, N. Akshay Kumar, M. Sivaguru, K. V. R. Keerthivaasan, B. Rohini Janaki, and R. Raffik, "Role of UAVs in innovating agriculture with future applications: A review," in *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAEECA)*, 2021, pp. 1–6.
- [2] A. S. Hanif, X. Han, and S.-H. Yu, "Independent control spraying system for UAV-based precise variable sprayer: A review," *Drones*, vol. 6, no. 12: 383, Nov 2022. [Online]. Available: <http://dx.doi.org/10.3390/drones6120383>
- [3] S. Zhao, Q. Wang, X. Fang, W. Liang, Y. Cao, C. Zhao, L. Li, C. Liu, and K. Wang, "Application and development of autonomous robots in concrete construction: Challenges and opportunities," *Drones*, vol. 6, no. 12: 424, Dec 2022. [Online]. Available: <http://dx.doi.org/10.3390/drones6120424>
- [4] S. Sreenath, H. Malik, N. Husnu, and K. Kalaichelavan, "Assessment and use of unmanned aerial vehicle for civil structural health monitoring," *Procedia Computer Science*, vol. 170, pp. 656–663, 2020, the 11th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050920306384>
- [5] S. Duan, X. Lu, H. Wu, H. Zhang, Z. Zhang, and G. Wang, "Unmanned aerial vehicle wireless power transfer system for long-distance transmission line patrol," in *2022 IEEE International Power Electronics and Application Conference and Exposition (PEAC)*, 2022, pp. 1324–1329.
- [6] S. H. Han, T. Rahim, and S. Y. Shin, "Detection of faults in solar panels using deep learning," in *2021 International Conference on Electronics, Information, and Communication (ICEIC)*, 2021, pp. 1–4.
- [7] X. Cao, "Safety analysis of UAV air transport operation," in *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 2021, pp. 203–206.
- [8] H. Eskandarpour and E. Boldsaikhan, "Last-mile drone delivery: Past, present, and future," *Drones*, vol. 7, no. 2: 77, Jan 2023. [Online]. Available: <http://dx.doi.org/10.3390/drones7020077>
- [9] C. Malang, P. Charoenvan, and R. Wudhikarn, "Implementation and critical factors of unmanned aerial vehicle (UAV) in warehouse management: A systematic literature review," *Drones*, vol. 7, no. 2: 80, Jan 2023. [Online]. Available: <http://dx.doi.org/10.3390/drones7020080>
- [10] Z. Xiaoning, "Analysis of military application of UAV swarm technology," in *2020 3rd International Conference on Unmanned Systems (ICUS)*, 2020, pp. 1200–1204.
- [11] P. Ramesh and J. M. L. Jeyan, "Comparative analysis of the impact of operating parameters on military and civil applications of mini unmanned aerial vehicle (UAV)," in *AIP conference proceedings*, vol. 2311, no. 1: 030034. AIP Publishing LLC, 2020.
- [12] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif, "Towards the unmanned aerial vehicles (UAVs): A comprehensive review," *Drones*, vol. 6, no. 6: 147, Jun 2022. [Online]. Available: <http://dx.doi.org/10.3390/drones6060147>
- [13] Y. Alghamdi, A. Munir, and H. M. La, "Architecture, classification, and applications of contemporary unmanned aerial vehicles," *IEEE Consumer Electronics Magazine*, vol. 10, no. 6, pp. 9–20, 2021.

- [14] N. Mohamed, J. Al-Jaroodi, I. Jawhar, A. Idries, and F. Mohammed, "Unmanned aerial vehicles applications in future smart cities," *Technological Forecasting and Social Change*, vol. 153: 119293, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0040162517314968>
- [15] R. Puchalski and W. Giernacki, "UAV fault detection methods, state-of-the-art," *Drones*, vol. 6, no. 11: 330, Oct 2022. [Online]. Available: <http://dx.doi.org/10.3390/drones6110330>
- [16] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: a survey," *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S235286481730247X>
- [17] H.-H. Xiao, W.-K. Yang, J. Hu, Y.-P. Zhang, L.-J. Jing, and Z.-Y. Chen, "Significance and methodology: Preprocessing the big data for machine learning on TBM performance," *Underground Space*, vol. 7, no. 4, pp. 680–701, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S246796742200006X>
- [18] A. Antonini, W. Guerra, V. Murali, T. Sayre-McCord, and S. Karaman, "The blackbird UAV dataset," *The International Journal of Robotics Research*, vol. 39, no. 10-11, pp. 1346–1364, 2020.
- [19] N. Mylonas, I. Malounas, S. Mouseti, E. Vali, B. Espejo-Garcia, and S. Fountas, "Eden library: A long-term database for storing agricultural multi-sensor datasets from UAV and proximal platforms," *Smart Agricultural Technology*, vol. 2, no. 100028, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2772375521000289>
- [20] P. Johnson, B. Ricker, and S. Harrison, "Volunteered drone imagery: Challenges and constraints to the development of an open shared image repository," in *Hawaii International Conference on System Sciences*, 2017.
- [21] S. Yural and C. Hacizade, "Sensor/actuator fault detection, isolation and accommodation applied to UAV model," *Journal of Aeronautics and Space Technologies*, vol. 9, no. 2, pp. 1–12, 2016.
- [22] M. Li, G. Li, and M. Zhong, "A data driven fault detection and isolation scheme for UAV flight control system," in *2016 35th Chinese Control Conference (CCC)*. IEEE, 2016, pp. 6778–6783.
- [23] B. Wang, X. Peng, M. Jiang, and D. Liu, "Real-time fault detection for UAV based on model acceleration engine," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 12, pp. 9505–9516, 2020.
- [24] M. H. M. Ghazali and W. Rahiman, "An investigation of the reliability of different types of sensors in the real-time vibration-based anomaly inspection in drone," *Sensors*, vol. 22, no. 16: 6015, Aug 2022. [Online]. Available: <http://dx.doi.org/10.3390/s22166015>
- [25] L. A. Al-Haddad and A. A. Jaber, "An intelligent fault diagnosis approach for multirotor UAVs based on deep neural network of multi-resolution transform features," *Drones*, vol. 7, no. 2: 82, Jan 2023. [Online]. Available: <http://dx.doi.org/10.3390/drones7020082>
- [26] A. Keipour, M. Mousaei, and S. Scherer, "ALFA: A dataset for UAV fault and anomaly detection," *The International Journal of Robotics Research*, vol. 40, no. 2-3, pp. 515–520, 2021. [Online]. Available: <https://doi.org/10.1177/0278364920966642>
- [27] STMicroelectronics, "STM32H742xI/G STM32H743xI/G, Datasheet - production data," 2022, accessed: 2023-01-24. [Online]. Available: <https://www.st.com/resource/en/datasheet/stm32h743xi.pdf>
- [28] InvenSense, "MPU-6500 Product Specification Revision 1.3," 2020, accessed: 2023-01-24. [Online]. Available: <http://invensense.wpgengineeringpowered.com/wp-content/uploads/2020/06/PS-MPU-6500A-01-v1.3.pdf>
- [29] Parrot, "Bebop-pro 3D modeling, All-in-one drone solution for 3D modeling," accessed: 2023-01-24. [Online]. Available: <https://www.parrot.com>
- [30] F. Chollet, *Deep Learning with Python*, 1st ed. New York, USA: Manning Publications Co., 2017.
- [31] H. Uyanık and M. Köseoğlu, "Performance evaluation of different window functions for audio fingerprint based audio search algorithm," in *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMISIT)*, 2020, pp. 1–4.
- [32] R. Puchalski, A. Bondyra, W. Giernacki, and Y. Zhang, "Actuator fault detection and isolation system for multirotor unmanned aerial vehicles," in *2022 26th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 2022, pp. 364–369.