

PRACTICAL ASPECTS OF DESIGN AND TESTING UNMANNED AERIAL VEHICLES

Patryk SZYWALSKI*, Andrzej WAINDOK*

*Faculty of Electrical Engineering, Automatic Control and Informatics Department of Electrical Engineering and Mechatronics, Opole University of Technology, ul. Prószkowska 76, 45-758 Opole, Poland

p.szywalski@doktorant.po.edu.pl, a.waindok@po.edu.pl

received 1 August 2019, revised 24 April 2020, accepted 28 April 2020

Abstract: A design of an unmanned aerial vehicle (UAV) construction, intended for autonomous flights in a group, was presented in this article. The design assumptions, practical implementation and results of the experiments were given. Some of the frame parts were made using 3D printing technology. It not only reduces the costs but also allows for better fitting of the covers to the electronics, which additionally protects them against shocks and dirt. The most difficult task was to develop the proper navigation system. Owing to high costs of precision positioning systems, common global positioning system (GPS) receivers were used. Their disadvantage is the floating position error. The original software was also described. It controls the device, allows performing autonomous flight along a pre-determined route, analyses all parameters of the drone and sends them in a real time to the operator. The tests of the system were carried out and presented in the article, as well.

Keywords: Unmanned Aerial Vehicles (UAV), Unmanned Aerial Systems (UAS), Navigation System, Trajectory Generation

1. INTRODUCTION

The four-engine flying devices, colloquially called quadcopters or drones, are very popular. In the literature, they can be found under the name remotely piloted aircraft systems (RPAS; Gómez et al, 2019), unmanned aerial systems (UAS) and unmanned aerial vehicles (UAV; Roseneia Rodrigues Santos de Melo et al, 2017). Typically, the term unmanned aerial vehicles (UAV) is used for one quadcopter and the term unmanned aerial system (UAS) is used for drones in number greater than one.

There is a growing interest in UAV observed. This is especially visible in the statistics of popularity of the keyword 'drone' in the Science Network (Thompson Reuters). This brought about 3,000 results for the past 15 years and the number has increased particularly rapidly in the past 4 years (María de Miguel Molina, 2018). At present, only the Swiss university ETH Zurich, the American company Intel and the Chinese company EHang Egret have drone network projects. These institutions have been successful in connecting the devices into one independent system.

At present, there are many publications focusing on flight simulations of different manoeuvres of one or many UAVs. These papers describe devices that can be freely scaled, combined with each other, and study the thesis based on the simulation (Nalapaneni et al, 2018; Pulvera et al, 2018). These devices have found their application in both industry (e.g. in geodesy, parcel transport and building protection or photography; Bonali et al, 2019; Choudhary et al, 2019; Fujimori et al, 2018) and amongst ordinary people who use them for recreational purposes. There are also solutions in which devices, connected with each other, are able to provide access to the Internet in certain crisis areas (Aghaeyan et al, 2015). The aim is also to integrate drones with autonomous cars (Khamseh et al, 2015; Kopiche et al, 2013).

The basic algorithm of stabilising an UAV in the overhang tends to stabilise three angles: pitch, roll and yaw (Fig. 1). Most control algorithms also have stabilisation of altitude, position and speed. To stabilise the pitch, roll and yaw angles, knowledge about their value is required. For this purpose, UAVs have a three-axis accelerometer and a three-axis gyroscope. Optionally, the device has a magnetometer to determine the direction along Earth's magnetic field (Ebeid et al, 2018).

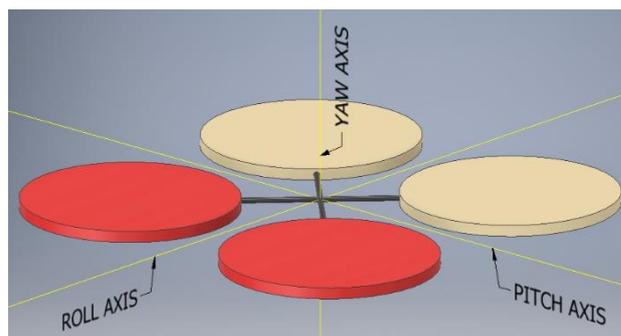


Fig. 1. Illustrative scheme of the UAV axis

The altitude parameter can be obtained from a barometric altitude sensor, laser sensor or other alternative navigation systems. The position of the device is taken from the navigation system, not necessarily global positioning system (GPS). Different alternative positioning systems, such as inertial navigation, video navigation or an indoor navigation system based on stationary ultrasonic beacons, are available. They are connected by radio interface in a license-free band (433 or 915/868 MHz; Sun, 2018).

Flight controllers with various processors can be found (Cetinsoy et al., 2012). The most popular is the STM 32 bit pro-

cessor in the F4 or F7 version. Ready modules (processor and basic sensors) are available from 20 Euro. The device can be used with a commercial software (often delivered with high-end equipment) or open-source software. The most popular of them are BetaFlight, CleanFlight, RaceFlight and iNav. The software allows programming the drone and configuring it for the flight.

In the article, practical aspects of design UAV are presented. The device is devoted for a drones' network and will be used in a testing of swarm algorithms based on nature such as swarm of bees, moths, ants, fishes and birds (Socha et al, 2008; Olivas et al, 2017; Luo et al, 2019). Construction of own device is necessary, because it is very difficult to purchase and modify a commercial UAV. The original construction allows fitting better the device for a specific application.

UAVs can be divided into several categories, for example, based on the number of engines, size or type of start (vertical or horizontal). However, the most common division is the number of engines: one, two, three, four or multi-engine drones can be distinguished (Szywalski et al., 2018, Puchala et al., 2015). In the

article, the construction of a four-engine device will be discussed.

2. STRUCTURE OF THE SYSTEM

In Figure 2 (and more detailed in Fig. 9) the structure of the system was shown. It is divided into three main parts:

- UAV (yellow) – this part presents one UAV, but it can be extended for an UAS
- Radio communication system (orange)
- Control and supervision software (green)

The UAV (or their collection) contains two STM32 processors, four drive units, sensors and a communication module. The whole system is mounted on a self-designed frame. Each processor has a pre-assigned task and priorities. In the first column (Fig. 2), priorities of the flight controller were shown. The most important task is to maintain the stability of the device, and the lowest priority is to maintain the right speed.

Unmanned Aerial Vehicles					Radio communication system	Control and Supervision Software				
STM F4	Data transfer UART	STM F2	Data transfer UART	Radio 433 MHz		Radio 433 MHz	Data transfer UART	M0	Data transfer UART	Computer
Stabilisation: pitch, roll and yaw	Data ↔	The autonomous flight algorithm	Data ↔	Data transmission	Data ↔	Data transmission	Data ↔	Intermediary module	Data ↔	Control panel
High stabilisation										
Position stabilisation										
Speed stabilisation										

Fig. 2. Structure of the system

The important element of the system is the radio communication module. It consists of radio modules mounted on each receiver in the network. For creating the network, 433-MHz RFM69 modules were used. The library RadioHead RFMxx was used to connect multiple clients by creating a network for data exchange. The system enables to exchange 60 bytes with a frequency of 20 Hz at a distance of up to 5 km (with appropriate antennas). The data transmission is encrypted with a 16-byte key. The created program is modular and allows adding drones (maximally 30) or flying out of operator's range (without a control panel).

The last module of the system is the control and supervision system controlled from the operator panel. To connect the computer to the network, a module that receives data from the radio network and sends it to the serial port on the computer was made. The software of the operator panel was created in MATLAB. This program processes data, draws relevant graphs and displays information.

3. DESIGN OF THE UAV

The goal of the project was to create a device with relatively high robustness to external forces, a flight time of minimum 15 min, weight up to 400 g, dimensions not exceeding 15 × cm × 15 cm × 15 cm and software enabling the device to be used in the network. In addition, it should be able to maintain integrity and availability during operations (if it is within operator's range). The

designed UAV should also be prepared for a navigation system that can determine the position of the device with an accuracy of 10 cm.

3.1. Description of electromechanical and electronic components used in the UAV

The challenge in the designing is the mechanical part, which should be characterised by high mechanical strength and low weight (in the presented case not exceeding 400 g). The central plate was milled from a carbon fibre-reinforced polymer (CFRP). The other elements were printed using a 3D printer. The print was made using the fused filament fabrication (FFF/FDM) method. It involves melting the material and laying it in thin paths, one next to another. The layers of plastic bond together, cooling down (Kownacki, 2016). For printing, the legs and two elements of the electronics housing the rubber material (TPU) and polycarbonate were used. The brass threads were embedded in the frame (Ferrarese et al, 2017), which was screwed using aluminium screws. The most important frame dimensions were given in Figure 3. The distance between the centre of mass (CoM) and axis of the propellers is also marked. The CoM is placed below the rotor line. The frame assembly is shown in Figure 4. Each element was sized and weighed. The mechanical part was designed in Inventor 2018.

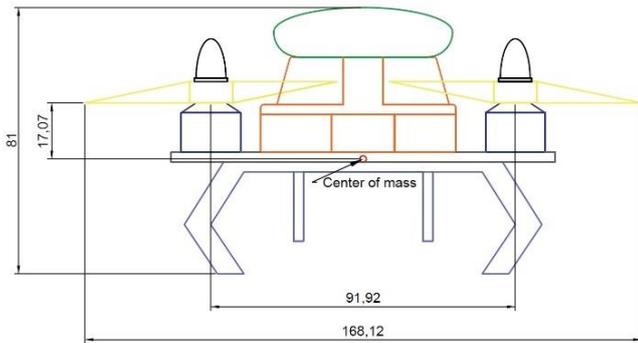


Fig. 3. Main dimensions of the drone and distance between the centre of mass and axis of the propellers (in mm)

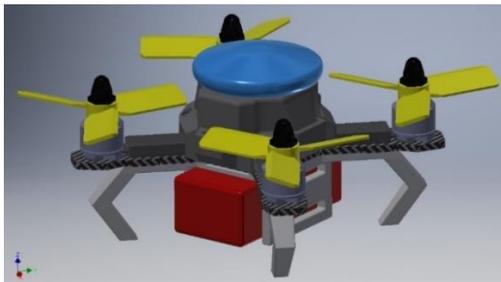


Fig. 4. Assembling of the frame elements

Table 1. Technical specification of the UAV controller

Controller
Processor: STM32F405 (168 MHz) and STM32F205RG (120 MHz)
Wi-Fi module: BCM43362 802.11b /G/N
Accelerometer/gyroscope: MPU6000 (SPI)
Radio module: RFM69HCW 433 MHz
Barometer: BMP280 (I2C)
Magnetometer: HMC5883 (UART)
GPS receiver: Ublox M8N (UART)
Blackbox: MicroSD card slot (SD/SDHC)
Beeper
Current/voltage sensor: 6-30 V/ 184 A (3-6S LiPo)
Step-down converter: 3.3 V, 4.5 V, 5 V – max. 3 A

The power unit of the device consists of four Readytosky 1306, 3100 KV engines and two pairs of propellers (2 x CW 3x3x3 and 2 x CCW 3x3x3). As the electronic speed controller (ESC) of the brushless direct-current motor (BLDC), the Little Bee 20A driver was selected (operates on the BiHeli software). The driver communicates with the controller using an OneShot125 protocol, which is characterised by eight times shorter pulses than the standard PWM protocol (from 125 to 250 μ s). It allows for eight times faster update of the PID control loop (loop update time of 250 μ s/4 kHz). The signal delay is also shorter: only 250 μ s instead of 2000 μ s. The OneShot125 protocol is needed for drones below 25 cm (diagonal distance of engines). MatekSys F405-CTR was chosen as the flight controller. It uses a powerful STM32F405 processor working at 168 MHz. In addition, the module has an inertial navigation system (IMU) sensor MPU6000, pressure sensor BMP280, blackbox – MicroSD (SD/SDHC), inverter for SBUS, PPM support, 6 x PWM, 5 x universal asynchronous receiver-transmitter (UART), SoftSerial, camera operation, operation programmable LEDs, beeper, current and voltage sensors and bat-

tery eliminator circuit (BEC) inverter. All parameters were included in Table 1.

The device was designed for using in a swarm of unmanned flying vehicles. Thus, the second module – Adafruit Feather WICED with the STM32F205RG processor – was added. The module will be used for communication with other UAVs and the operator. Its second task is to calculate the flight trajectory in a real time. The radio module – Adafruit FeatherWing LoRa RFM69 433MHz – was connected to it. Both modules can be mounted into a ‘sandwich’, creating one controller (Figs. 7 and 8).

The MatekSys 405-CTR module communicates with Adafruit Feather WICED via the serial port using the MSP protocol (Multi-wii Serial Protocol). The protocol works based on the client server. In this case, the master controller is Feather WICED Adafruit with the STM32F405 processor on board. It sends queries for specific parameters, calculates the predicted trajectory based on this and sends coordinates and other parameters (e.g. flight speed) to the flight controller. A detailed diagram of the system and the division of specific tasks between the single modules is shown in Figures 2 and 9. In Figures 5 and 6, the pictures of completed UAV are shown.



Fig. 5. Final construction – view from the side



Fig. 6. Final construction – view from the skew

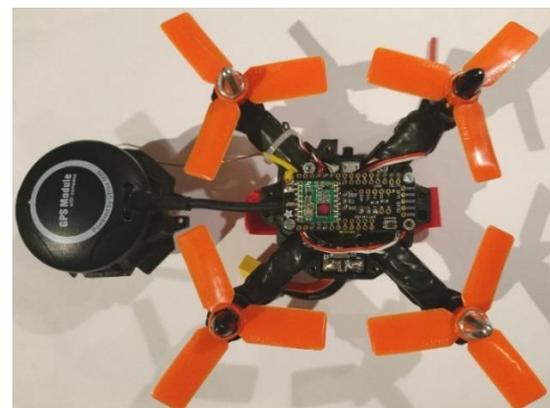


Fig. 7. Electronics assembly – view from the top

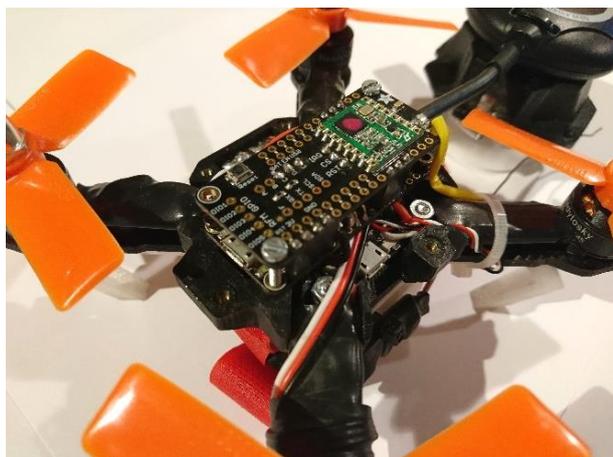


Fig. 8. Electronics assembly – view from the skew

Table 2. Technical specification of the UAV

Quadcopter
Size without propellers: 12 cm × 12 cm
Size with propellers: 15.5 cm × 15.5 cm
Battery: Lithium polymer 11.1 V 1550 mAh
Type of propellers: 2xCW 3x3x3, 2xCCW 3x3x3
Weight: 365 gram
Motor BLDC: Readytosky 1306, 3100KV, 58.5W
ECS: Little Bee 20A BIHeli
Frame: class 130 <ul style="list-style-type: none"> • central plate: CFRP, • electronics housing: polycarbonate, • legs: thermoplastic polyurethane (TPU).

Specifications of the device were included in Table 2. The beeper was used in order to find the device after an unplanned or unsuccessful landing. For localisation, the Ublox M8N GPS receiver was used. The GPS receiver and magnetometer were placed on the top of the device. It is possible to find UAVs with a magnetometer built inside the machine, but in this case, motors

or power supply often disturbs the measurement. Although the sensor is pulled out of the device, it is necessary to cancel the electromagnetic field generated by the motors and other components.

3.2. Schematic of electrical part

In Figure 9, the outline of electrical part of the entire system was shown. A yellow line surrounded the quadcopter block. In the centre, a flight controller is visible. Owing to the dynamics of the drone, the engines rotate in pairs in the opposite direction. Two types of propellers (clockwise [CW] and counterclockwise [CCW]) were used. They were assembled in such a way that the thrust is directed downwards. In the project, four three-phase BLDC motors were used. The device is powered by a 3-cell, 11.1 voltage polymer-lithium battery. Their big advantage is a very high discharge current. The battery allows power consumption up to 1 kW. In order to eliminate a voltage drops for the current peaks, a 470 μF, 25 V capacitor was added to the power connector. The F405-CTR module is equipped with step-down converter from 11.1 V to 3.3, 4.5 and 5 V. It serves as a source for powering the other modules (e.g. GPS, radio module and beeper). The GPS receiver and radio module communicates with the flight controller using the I2C bus. The magnetometer (mounted in the same housing as GPS) uses the SPI protocol.

A green line surrounds the operator supervision block. It consists of two modules (responsible for communication), a computer, original software (shown in Fig. 2), a safety button and a battery (980 mAh, 3.3 V). The battery is constantly recharged when the system is powered from the computer's USB port. Its main task is to maintain the battery voltage during a sudden power failure on the computer (e.g. discharge or power failure). This solution allows sending a landing order, even without the operator panel. For a protection, the system includes a safety button that is connected directly to the radio module. After pressing the button, regardless of whether the computer is on and/or other commands, the radio module sends an immediate order to land. The third block – radio communication – is surrounded with orange line.

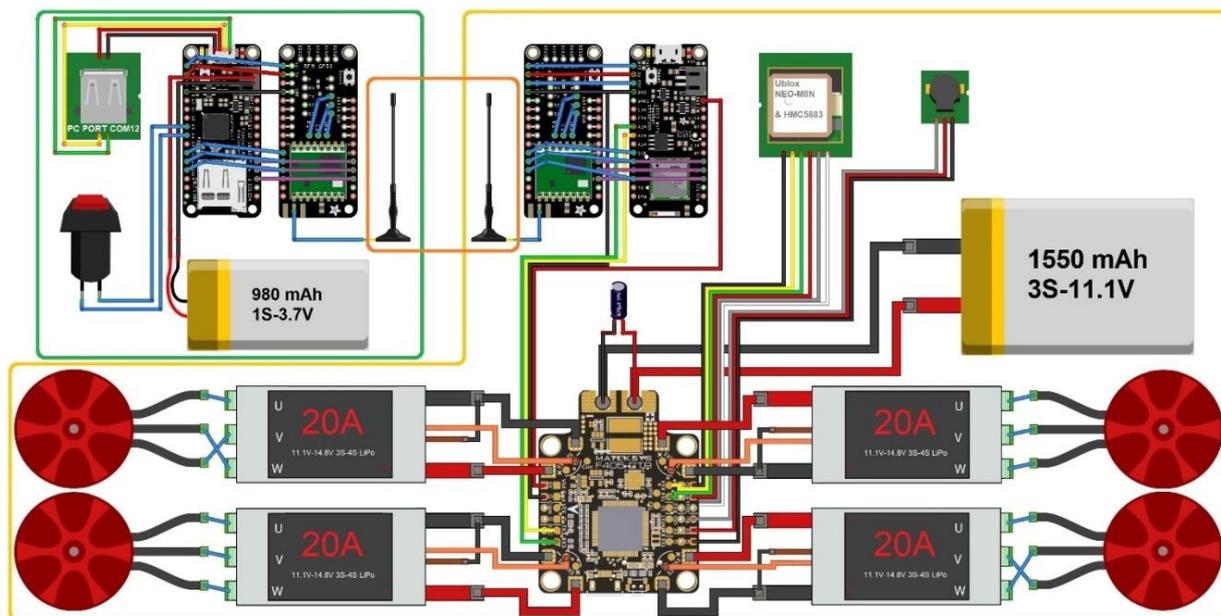


Fig. 9. Outline of the electric part

3.3. Navigation system

For autonomous flight, an unmanned flying unit must know its position in three-dimensional space. For this reason, it is necessary to use navigation systems. The most popular and easiest to use is GPS. Its biggest advantage is the possibility of using it all over the globe. To determine the position in three-dimensional space, it is necessary to connect the receiver with a minimum of three satellites. The accuracy of the position increases with the number of connected satellites. The system has been configured forcing a connection with a minimum of six satellites. A smaller number of satellites do not allow starting in the GPS-assisted mode. At present, the most popular receiver of the GPS signal is Ublox NEO-M8N. Its disadvantage is accuracy not exceeding 2.5 m in relation to latitude/longitude and about 5 m in relation to altitude.

In Figures 10 and 11, the static characteristics of the tested receiver were shown. The experiment was performed using the iNav software. The measurement was performed in the following way: the receiver was turned on in an unbuilt area and the maximum displacement during 10 min and during 10 s was measured. The experiment was started only after establishing communication with a minimum of 12 satellites. By converting the longitude and latitude values to meters, it is assumed that the Earth is a sphere, not an ellipsoid, and that the errors in the shipments tested (not exceeding 10 km) are not large enough to affect the results (Szywalski et al, 2018).

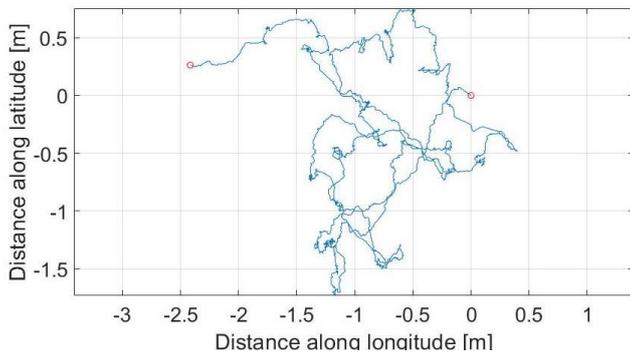


Fig. 10. Static 2D characteristics of GPS receiver (10 min of measuring)

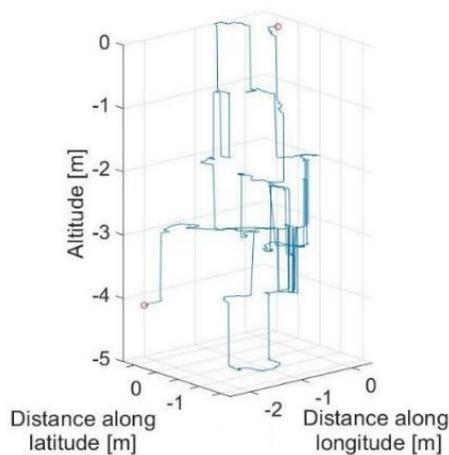


Fig. 11. Static 3D characteristics of GPS receiver (10 min of measuring)

In order to evaluate the navigation system, an error for a sample of 10 min and 10 s was calculated. The value of 10 min is important for long flights. It indicates how far the destination can be moved. The deviation was also calculated for 10 s period because, in this time, the device passes between the neighbouring points. Detailed analysis of the results showed that the maximum difference between the measurement (longitude and latitude) in the sample n and $n + 1$ was not greater than 14 cm, and the average was 4 mm. This is important because, although the accuracy of the system is several metres, the measurement value does not change significantly between the two measurement points. The results related to the displacement from the starting point $P_0 = (0,0,0)$ are given in Table 3.

The altitude value is characterised by a relatively larger error compared to longitude and latitude values. This error can be minimised with using an additional barometric height sensor. The sensor uses the dependence of the pressure verso the height according to the formula (Zhu et al., 2013):

$$h = \frac{T_o \left(\left(\frac{p}{P_o} \right)^{\frac{R \cdot L_o}{g_o \cdot M}} - 1 \right)}{L_o} \quad (1)$$

where h is the altitude above sea level (in m); T_o is the standard temperature above sea level 288.15 K; L_o is the constant of temperature change, 0.0065 K/m; p is the measured pressure in hPa; P_o is the static pressure 1013.25 hPa; g_o is the acceleration of gravity, 9.80665 m/s²; M is the molar mass of air, 0.0289644 kg/mol; and R is the gas constant, 8.31432 J/(mol*K)

The barometric height sensor is able to measure several dozen centimetres with the assumption that the sensor is not affected by any external factors: air turbulence generated by the drive units, wind or temperature fluctuations. The sensor must be mounted at a suiTab, placed away from the elements that can heat up during the work. In the presented construction, it was additionally wrapped with a sponge.

Tab.3. Static measurement results for the Ublox NEO-M8N GPS

Error type		Distance along latitude (m)	Distance along longitude (m)	Altitude (m)
Measurement for 10 min	Average distance from P0	0.616	0.903	2.955
	Span of the measurement	2.482	2.816	5
	Max distance from P0 in R3	5.553		
Measurement for 10 s	Average distance from P0	0.576	0.465	0.87
	Span of the measurement	0.99	1.057	1
	Max distance from P0 in R3	1.435		

The overworked control algorithm takes into account the values taken from the GPS and from the barometric altitude sensor. In addition, in order to increase the accuracy, a radar could be used (radar altimeter and electronic altimeter). By means of appropriate fusion algorithms, for example, Kalman, the height can

be determined with an accuracy of a few centimetres (Cho et al, 2011, Cai et al, 2008). An another solution is Inertial Navigation System (INS), Vision Navigation System (VNS) or INS/VNS fused by a Kalman filter (Huang et al, 2018; Stančić et al, 2010; Cechowicz et al, 2017).

3.4. Open-Source flight controller software

The control algorithm was made based on an open-source flight software. Before starting, it was necessary to properly connect, configure and calibrate the sensors, as well as to select the flight mode. At present, it is possible to find many open-source programs to control a quadcopter with a STM32 processor. The software can be divided according to the function:

- CleanFlight – is characterised by a stable and easy-to-use code. It is updated, on an average, once a quarter. At present, it supports STM32 processors – F1, F3, F4 and F7.
- BetaFlight – experimental software. On an average, new features appear once a week. It supports STM32 processors – F1, F3, F4 and F7.
- RaceFlight – created especially for racing drones. To increase the operating frequency of the control loop, some unnecessary stabilisation functions and GPS module were removed. It supports STM32 – F4 and F7 processors.
- iNav – is a software created for GPS flights and Return to Home (RTH) functions. It is used mainly for recording movies from the air and GPS-assisted flights. It supports STM32 – F3, F4 and F7 processors.

Cleanflight, BetaFlight and iNav can be used on multirotor aircrafts and fixed-wing aircraft. They support a variety of shapes and motor counts not limited to quadcopters, hexacopters, octocopters, tricopters and planes. The iNav software was selected according to the intended use of the device. An advantage is a big pressure of the creators on a sTab. and precise flight using the navigation systems.

A useful function of the iNav software is the RTH and Waypoint (WP) algorithm. In the presented solution, the RTH functions is used as a safeguard – for example, loss of communication or other errors (e.g. from navigation systems). The accuracy of the function has been tested and the device actually lands at a maximum distance of 2.5 m from the starting point. WP functions were used to control the device. The STM32F2 driver calculates the direction of the flight and sends it to the flight controller in the form of point coordinates and speed of reaching it. It is important that the target point should be not too near to the actual position of the device – in such a case, the system recognise the point as completed. This is due to the accuracy of GPS. In the presented construction, the version of iNav 2.1.3 and the Firmware for MATEK405 2.0.1 were used.

3.5. Planning and generating trajectories

Original software was created to operate the drone, allowing generating a trajectory, sending it to the device and supervising the quadcopter during the flight. The whole system consists of a quadcopter, a communication system and a PC-class computer. The final trajectory consists of several shapes (line, circle and helix; Szywalski, 2017). It was assumed that the user should define the characteristic points of each partial shape and define the type between the points. It is possible to create a shape and

then duplicate it. Owing to the inaccuracy of navigation systems, each result has an error, so it is impossible for the device to reach a certain point directly. Therefore, around each point a region was defined – it has a sphere shape characterised by a diameter of 1.5 m. The diameter parameter can be modified depending on the flight accuracy. When the distance between the main points is bigger than the declared accuracy, the program will automatically calculate intermediate points in such a way that the defined shapes are maintained. The idea of the algorithm is to generate a virtual tunnel in which the device should move. When the UAV reaches the defined zone, it is assumed that the point was reached and the device can fly to the next one.

The arrows in Figure 12 indicate the order of the points (flight direction). The flight path is marked in green. The cross section of the track is a circle. For this example, 1 m was accepted as the distance between the intermediate points. Increasing this parameter reduces the accuracy of the flight. By defining the distance (between the intermediate points) and the regions around them, the accuracy of the GPS should be taken into account. The software allows re-calling predefined shapes, scaling them and saving.

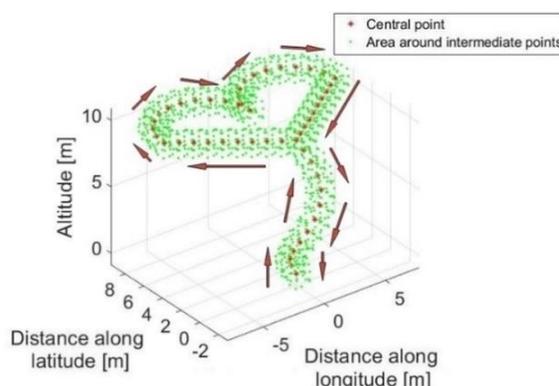


Fig. 12. Trajectory in the shape of a heart

3.6. Operator panel

In order to test the UAV work for different trajectories, the original software made in MATLAB was used. The operator panel is shown in Figure 13. The software allows generating or importing a flight trajectory from an external, previously created file. The whole software works in a real time and displays following parameters: start position, current position, distance from the starting point, number of visible satellites of the GPS system, statistics of data transmission, battery voltage, instant current consumption and energy consumption, and draws a displacement in three and two axes. The 2D drawing has additionally a graph, which shows how the device is turned to the north. The data can be saved to an external file and replayed.

The software was also created by taking into account the flight safety and to predict the discharge of the battery. To detect early errors in the data transmission between the operator and the drone, all of the parameters are analysed. In the normal operating mode between the operator and the drone, 15 device parameters are sent in 56 bytes. The system evaluates the correctness of the parameter based on predefined rules (Table 3). When the parameter is incorrect, dm is 1. Then the sum of the incorrect parameters is counted and multiplied by their predicted length in bytes. Missing data packet (all n parameters) in 100 ms time increases

the t_{err} counter. As a result, the system calculates the error of false or missing bytes in relation to the predicted bytes in data transmission according to the following formula:

$$err = \frac{\sum_{i=1}^{nd} (\sum_{j=1}^n (dm_j \cdot dp_j)) + (t_{err} \cdot dp_size)}{nd \cdot dp_size \cdot 0.01} \quad (2)$$

nd is the number of demanded packets, n is the number of sent parameters (in our case 15), dm is the data message (1 represents error and 0 is correct parameter), dp_j is the number of bytes of j -th parameter, dp_size is the data packet size (normally 56 byte) and t_{err} is the counter for missing data packages.

The resulting ratio is expressed as a percentage of how much data has been lost. For a value above 20%, the device returns autonomously to the starting point. In the tests, the parameter was below 3%, which is within the limits of acceptance.

In Table 4, a list of all parameters that are sent from the device to the operator is presented. For each parameter, its number of bytes and boundary conditions were defined. If either condition is not met, the system sends a message consisting of zeros and (on the last byte) 'e', for example, '00000e' for a parameter of 6 bytes. Thus, the system knows that the parameter was read incorrectly or changed when sending to the operator. To minimise the data package, parameters GPS_LON and GPS_LAT are reduced from 9 to 6 bytes. The system has the first 3 bytes permanently assigned (they have a constant value). The result is the reduction of the zone (to 7.1 km × 11.1 km) in which the device can move. In the M1-M4 parameters, additional messages were programmed:

- 0–50: the messages inform about the motor arming,
- 50–950: motor speed,
- 950– 999: motor error messages.

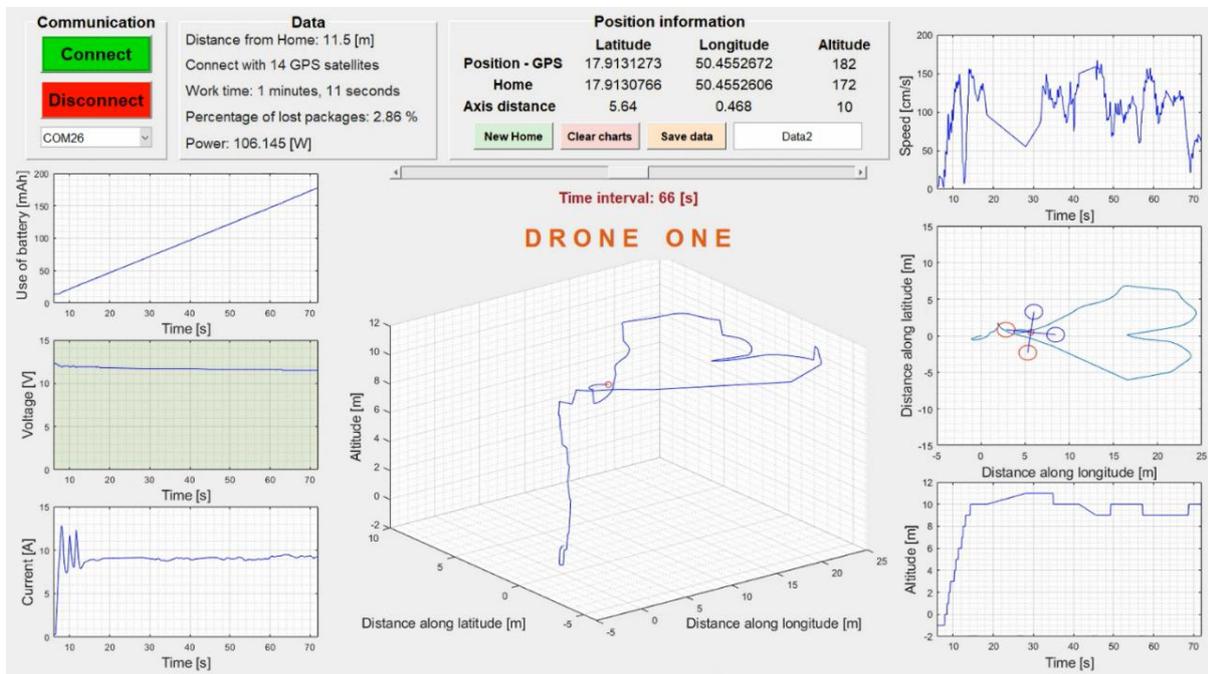


Fig. 13. Operator panel

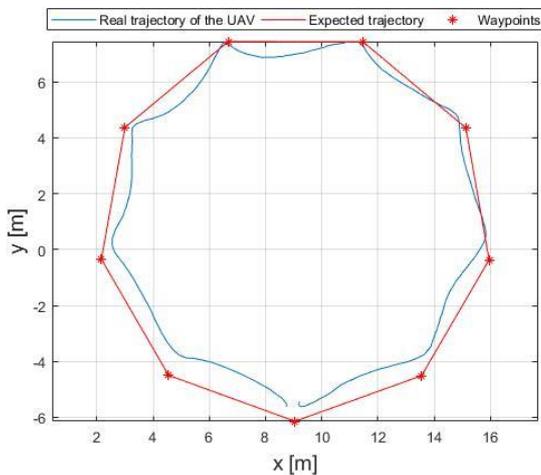


Fig. 14. Flight along a circle with a diameter of 14 m (flight speed: 75 cm/s; wind speed: 4 m/s): the average path error is 0.4 m and the maximum path error is 0.96 m

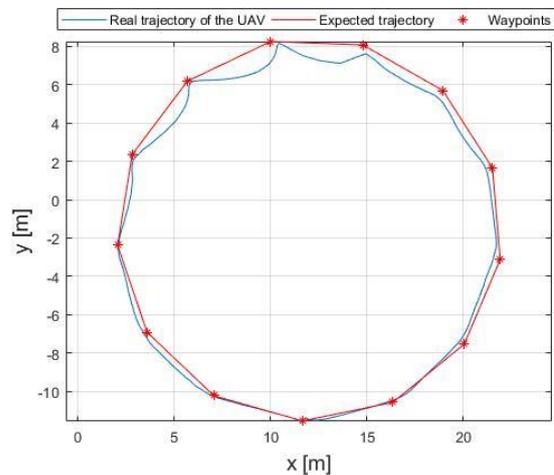


Fig. 15. Flight along a circle with a diameter of 20 m (flight speed: 150 cm/s; wind speed: 4 m/s): the average path error is 0.28 m and the maximum path error is 1.13 m

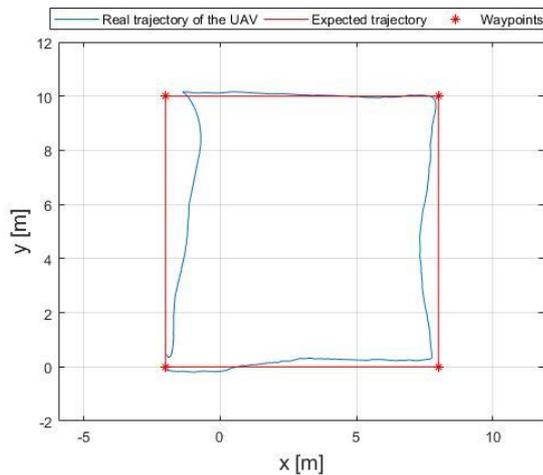


Fig. 16. Flight along a rectangle with a side length of 10 m (flight speed: 75 cm/s; wind speed: 4 m/s): the average path error is 0.3 m and the maximum path error is 1.29 m

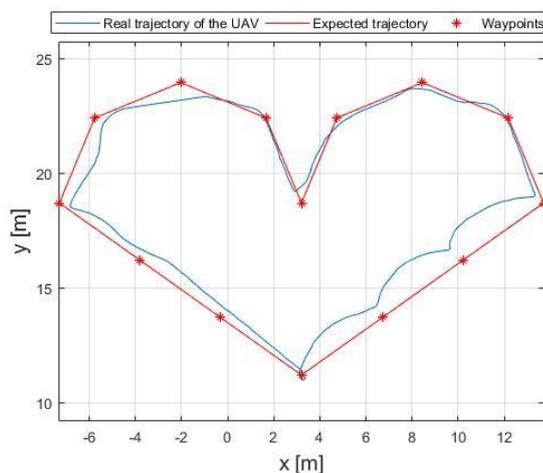


Fig. 17. Flight along a heart trajectory (flight speed: 150 cm/s; wind speed: 4 m/s): the average path error is 0.38 m and the maximum path error is 1.36 m

The energy consumption is calculated, as well. On the basis of these tests, it is possible to predict the discharge of the battery. There are many publications devoted to this issue. One of the most effective solutions includes neural network (Souza et al, 2016). The graphs on the left of Figure 13 show (from above): energy consumption in mAh, battery voltage and instant current consumption. The energy consumed is expressed in mAh, which allows direct comparison of the value to the nominal capacity of the battery. The second graph shows the voltage on the battery. If the voltage drops below 10.8 V (3.6 V on one cell), the background will change to yellow, and if it drops below 9.6 V (3.2 V on one cell), background will change to red indicating battery discharge and immediate landing. The green background of the graph indicates the correct battery voltage. The bottom chart shows the instantaneous current consumption. On its basis, the power consumption of the device can be calculated (in the presented case, it is 106.145 W – the device was in the overhang). During the programming process, the device consumes about 0.7 W, and when the battery and ECS drivers are connected, it consumes 3.4 W.

In the centre of Figure 13, a graph with the UAV displacement in three axes is shown. It should be noted that the charts have been drawn relative to the local coordinate system where the point

(0, 0, 0) is the place where the engines are armed – the device is starting. On the right side, three graphs are present – the data from the GPS receiver. The top plot shows the current speed of the device – in our case, it is 100 cm/s. The value is characterised by an error, which is due to the low priority of the control algorithm. An accurate speed measurement can be performed based on the visual and inertial information without GPS. In this method, no map and no artificial landmark of the environment are required – only the off-the-shelf onboard sensors in a multicopter including a low-cost inertial measurement unit (IMU), a downward-looking monocular camera and an ultrasonic range finder facing downwards are needed to constitute the vision motion constraint (Deng et al, 2018). The middle graph shows the device displacement in 2D. The current position of the device is shown by the graphics of a four-engine drone. The front of the device is marked in red. This graphic rotates according to the actual rotation of the device. The last graph shows the altitude parameter.

Table 4. List of device parameters and their length in bytes

	Parameter	Number of bytes	Rules causing an error
1	GPS_LON	6	Size > 6 GPS_LON > 180000000 GPS_LON < 178000000
2	GPS_LAT	6	Size > 6 GPS_LAT > 505000000 GPS_LAT < 503000000
3	GPS_ALT	3	Size > 3 GPS_ALT > 999 GPS_ALT < 120
4	GPS_numSat	2	Size > 2 GPS_numSat > 20 GPS_numSat < 0
5	angx	4	Size > 4 angx > 180 angx < -180
6	angy	4	Size > 4 angy > 180 angy < -180
7	heading	4	Size > 4 heading <= 0 heading >= 360
8	vbat	3	Size > 3 vbat > 200 heading < 0
9	mAhDrawn	4	Size > 4 mAhDrawn > 2000 mAhDrawn < 0
10	amperage	4	Size > 4 amperage > 9999 amperage < 0
11	GPS_speed	4	Size > 4 GPS_speed > 9999 GPS_speed < 0
12	M1	3	Size > 3 M1 > 999 M1 < 0
13	M2	3	Size > 3 M2 > 999 M2 < 0
14	M3	3	Size > 3 M3 > 999 M3 < 0
15	M4	3	Size > 3 M4 > 999 M4 < 0

4. SUMMARY

The autonomous flight tests were made for different trajectory shapes. Owing to clarity, the demanded and measured trajectories were shown in 2D plane (Figs. 14–17). The differences between them do not exceed the GPS error, which confirms the correctness of the UAV construction and control algorithms. The movies, showing the completed operator panel during the work for tested trajectories are available on YouTube under links given in References 32 (circle with 14-m diameter), 33 (circle with 20-m diameter), 34 (rectangle) and 35 (heart trajectory).

The developed device can flight up to 16 min in the autonomous mode. The structure of the system is scalable. The test shows that the UAV works sTab. and flights in autonomous mode with or without the operator panel. In addition, the RTH and WP algorithms work correctly.

The average path error (deviation between demanded and real flight path) during the flight did not exceed 0.4 m, whereas the maximum path error was below 1.4 m. It was found that increasing the number of intermediate points reduces the error value. Considering the flight using GPS, the obtained error values are satisfactory. The use of more accurate navigation systems (e.g. local), which will be tested in further work, should reduce these errors.

The mechanical elements made using 3D print technology met the expectations. The housing withstood the tests and protected the electronic part of the device. All assumptions were achieved.

The system was made for further studies related to the optimization of the flight. The presented UAV will be used in testing of swarms algorithms.

REFERENCES

1. **Aghaeeyan A., Abdollahi F., Talebi H.A.**, (2015), UAV–UGVs cooperation: With a moving center based trajectory, *Robotics and Autonomous Systems*, 63, Part 1, 1-9.
2. **Bonali F.L., Tibaldi A., Marchese F., Fallati L., Russo E., Corselli C., Savini A.**, (2019), UAV-based surveying in volcano-tectonics: An example from the Iceland rift, *Journal of Structural Geology*, 121, 46-64.
3. **Cai G., Feng L., Chen B., Lee T.H.**, (2008), Systematic design methodology and construction of UAV helicopters, *Mechatronics* 18, 545–558.
4. **Cechowicz R.**, (2017), Bias drift estimation for mems gyroscope used in inertial navigation, *Acta Mechanica et Automatica*, 11(2), 104-110.
5. **Cetinsoy E., Dikyar S., Hancer C., Oner K.T., Sirimoglu E., Unel M., Aksit M.F.**, (2012), Design and construction of a novel quad tilting UAV, *Mechatronics* 22, 723–745.
6. **Cho A., Kang Y.S., Park B., Yoo Ch.S., Koo S.O.**, (2011), Altitude Integration of Radar Altimeter and GPS/INS for Automatic Takeoff and Landing of a UAV, *2011 11th International Conference on Control, Automation and Systems*, Gyeonggi-do, Korea, 1429-1432.
7. **Choudhary G., Sharma V., You I.**, (2019), Sustainable and secure trajectories for the military Internet of Drones (IoD) through an efficient Medium Access Control (MAC) protocol, *Computers & Electrical Engineering*, 74, 59-73.
8. **Deng H., Arif U., Fu Q., Xi Z., Quan Q., Cai K.**, (2018), Visual-inertial estimation of velocity for multicopters based on vision motion constraint, *Robotics and Autonomous Systems*, 107, 262-279.
9. **Ebeid E., Skriver M., Husum K., Jensen K., Pagh U.**, (2018), A Survey of Open-Source UAV Flight Controllers and Flight Simulators, *Microprocessors and Microsystems*, 61, 11-20.
10. **Ferrarese G.**, (2017), **Bandwidth Assessment for MultiRotor UAVs**, *Acta Mechanica et Automatica*, 11(2), 150-153.
11. **Fujimori A., Ukigai Y., Santoki A., Oh-hara S.**, (2018), Autonomous flight control system of quadrotor and its application to formation control with mobile robot. *IFAC-PapersOnLine*, 51(22), 343-347.
12. **Gómez A., Rodríguez A., Sanchez C., Luis G., Hernández C., Cuerno R.**, (2019), Remotely Piloted Aircraft Systems conceptual design methodology based on factor analysis, *Aerospace Science and Technology*, 90, 368-387.
13. <https://www.youtube.com/watch?v=4rh5Z1fhzq4&feature=youtu.be> (access on 23.12.2019).
14. <https://www.youtube.com/watch?v=4WOrWoNT-bM&feature=youtu.be> (access on 23.12.2019).
15. <https://www.youtube.com/watch?v=eJ9QHfDsagQ&feature=youtu.be> (access on 23.12.2019).
16. <https://www.youtube.com/watch?v=tq4ihl6fRDg&feature=youtu.be> (access on 23.12.2019).
17. **Huang L., Song J., Zhang Ch., Cai G.**, (2018), Design and performance analysis of landmark-based INS/Vision Navigation System for UAV, *Optik*, 172, 484-493.
18. **Khamseh H.B., Janabi-Sharifi F., Abdessameud A.**, (2018), Aerial manipulation—A literature survey, *Robotics and Autonomous Systems*, 107, 221-235.
19. **Kopichev M., Ignatiev K., Putov A.**, (2013), Autonomous Control and Stabilization System for Unmanned Aerial Vehicles, *IFAC Proceedings Volumes*, 46(30), 240-243.
20. **Kownacki C.**, (2016), Multi-UAV Flight on the Basis of Virtual Structure Combined with Behavioral Approach, *Acta Mechanica et Automatica*, 10(2), 92-99.
21. **Luo Q., Yang X., Zhou Y.**, (2019). Nature-inspired approach: An enhanced moth swarm algorithm for global optimization, *Mathematics and Computers in Simulation*, 159, 57-92.
22. **María de Miguel Molina, Virginia Santamarina Campos, M. Ángeles Carabal Montagud, Blanca de Miguel Molina**, (2018), Ethics for civil indoor drones: A qualitative analysis, *International Journal of Micro Air Vehicles*, 10(4), 340–351.
23. **Nallapaneni Manoj Kumara, Sudhakar K., Samykano M., Jayaseelan V.**, (2018), On the technologies empowering drones for intelligent monitoring of solar photovoltaic power plants, *International Conference on Robotics and Smart Manufacturing (RoSMA2018)*, Procedia Computer Science, 133, 585–593.
24. **Olivas F., Valdez F., Castillo O., González C.I., Martínez G.E., Melin P.**, (2017), Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems, *Appl. Soft Comput.*, 74-87.
25. **Puchała K., Szymczyk E., Jachimowicz J.**, (2015), FEM design of composite – metal joint for bearing failure analysis, *Przegląd Mechaniczny*, 33 – 41.
26. **Pulvera A., Weib R.**, (2018), Optimizing the spatial location of medical drones, *Applied Geography*, 90, 9–16.
27. **Roseneia Rodrigues Santos de Melo, Dayana B.C., Juliana Sampaio Álvares, Irizarry J.**, (2017), Applicability of unmanned aerial system (UAS) for safety inspection on construction sites, *Safety Science*, 98, 174-185.
28. **Socha K., Dorigo M.**, (2008), Ant colony optimization for continuous domains, *European Journal of Operational Research*, 1155-1173.
29. **Souza D., Pinto V., Nascimento L., Torres J., Gomes J., Sa-Junior J., Sa-Junior J., Almeida R.**, (2016), Battery Discharge forecast applied in Unmanned Aerial Vehicle, *Przegląd Elektrotechniczny* 02/2016, 185-192.
30. **Stančić R. Graovac S.**, (2010), The integration of strap-down INS and GPS based on adaptive error damping, *Robotics and Autonomous Systems*, 58(10), 1117-1129.
31. **Sun J., Li B., Wen Ch.Y., Chen Ch.K.**, (2018), Design and implementation of a real-time hardware-in-the-loop testing platform for a dual-rotor tail-sitter unmanned aerial vehicle, *Mechatronics* 56, 1–15.
32. **Szywalski P.**, (2017), Design of the autonomous flight algorithm for Unmanned Aerial System, Opole, 4-61
33. **Szywalski P., Waindok A.**, (2018), Analysis of the quadcopter class 130 frame deformation made with using 3D printing technology, *Przegląd Mechaniczny*, 39-44.
34. **Szywalski P., Wajnert D.**, (2018), Possibility Analysis of the Location Measurement by Using the GPS Receiver and Barometric Altimeter, *Pomiary Automatyka Robotyka*, 33-39.
35. **Zhu W., Dong Y., Wang G., Qiao Z., Gao Z.**, (2013), High-precision Barometric Altitude Measurement Method and Technology, *2013 IEEE International Conference on Information and Automation (ICIA)*, 430-435.