Pol J Aviat Med Bioeng Psychol 2022; 28(1): 5-21
DOI: 10.13174/pjambp.28.03.2025.01

ORIGINAL ARTICLE

DESIGN AND TESTING OF A DRONE PILOT TRAINING SYSTEM USING MIXED REALITY

Antoni KOPYT[1], Cyprian FELENCZAK[1]
1 Warsaw University of Technology, Faculty of Power and Aeronautical Engineering, Warsaw, Poland

Source of support: This work was supported by Boeing company, students grant, "Development of an advanced, innovative tool to improve a UAV's operator training process using Augmented Reality", grant no. 540060100235, 2024

Author's address: A. Kopyt, 22/5 Nowowiejska Street, 00-665 Warsaw, Poland, e-mail: antoni.kopyt@pw.edu.pl

Introduction: Unmanned Aerial Vehicles (UAVs), commonly known as drones, have emerged as versatile tools with a wide range of applications, from surveillance and aerial photography to disaster management and agricultural monitoring. The increasing ubiquity of drones in various industries has driven a growing demand for skilled drone pilots who can operate these vehicles safely and effectively. The present pilots' training methods are based on the subjective assessment and knowledge of the instructor. Despite the instructor's perfection in the domain, the lack of objective parameters and repetitiveness of the assessment makes the training less efficient than it could be.

Objectives: The design and development of an innovative drone pilot training system is therefore presented, with the aim of making training and examinations repeatable and objective procedures. Using the mixed reality head-mounted display HoloLens 2, a customizable training environment has been created, along with a scoring algorithm that measures and automatically assesses pilots' performance.

Results: The system was assessed, initially within a simulated environment, and subsequently through real-world flight tests. The set of in-flight tests with full equipment has been developed at the Aviation Research Center (Przasnysz airfield), which is a part of Warsaw University of Technology.

Conclusions: It was concluded that the system offers a promising outlook to address the evolving and growing demands of the UAV industry. The objective assessment of a pilot's performance is a valuable tool for the operator and instructor during the training process.

Keywords: UAV, drones, mixed reality, HoloLens 2, training, pilot's assessment

Cite this article: **Kopyt A, Felenczak C. Design and Testing of a Drone Pilot Training System Using Mixed Reality. Pol J Aviat Med Bioeng Psychol 2022; 28(1): 5-21. DOI: 10.13174/pjambp.28.03.2025.01**

INTRODUCTION

This paper focuses on the training and certification of UAVs pilots in visual line-of-sight (VLOS) flight as required by aviation authorities, and more specifically on their practical part. The main weak point of the training courses that exist today is the subjectivity and lack of repeatability of the flight evaluation, as it is done visually by the instructor sitting next to the pilot-student [2,3]. During the practice training, the UAV operator is performing the set of maneuvers. Each manure has its own criterion to pass. However, the assessment of the training is based on the subjective opinion of the instructor. Such a solution creates a space for small errors due to this kind of assessment. The training and examination would be more efficient if both the instructor and the operator could base it on real data acquired from the UAV flight parameters. To solve this issue, a developed system is proposed that fulfils two main functionalities:

– display in real time to the pilot information about the exercise, errors made, and flight parameters of the UAV,
– generate an objective evaluation after the exercise to allow comparison between successive flights by the same pilot or between different pilots and generate an exercise report that allows the instructor to have more insight.

In order to fulfill the first functionality, it was assumed that the project would use HoloLens 2 (HL2) mixed reality goggles [7,8]. They should receive information about the set exercise and the UAV's flight parameters and then display the virtual environment to the pilot in conjunction with the real one, and the data as a head-up display (HUD). This approach is similar to the use of flight simulators in aviation, incorporating external measurements to obtain information concerning human factors and the operator's assessment [9].

The second functionality will be fulfilled by a suitably designed scoring algorithm, which will calculate a score based on the differences between the tasks set in the exercise and the actual flight. It will be placed after the flight in a report generated by the instructor. The overarching assumption of the project is that the system is intended to provide support to the instructor, rather than entirely replacing them. In this concept, emphasis was placed on high configurability and relative simplicity of use. Example exercises were written with the intention of testing and demonstrating the functionality and capabilities of the system. The design was aimed at drones based on the ArduPilot autopilot, due to their popularity among amateur builders and pilots [6].

METHODS

System design

Fig. 1 shows a schematic of the designed drone pilot training system. As can be seen in the figure, it consists of two main components in the form of a ground station and a HoloLens 2 (Microsoft, USA) device, and a drone. The UAV is independent of the designed system and is not modified in any way. It only needs to transmit data about its position and orientation using the MAVLink protocol [3].

The UAV's telemetry data is received by a radio modem (or a different drone communication device) hooked up to a computer running the ground station software. Communication is then established between the computer and the HoloLens 2 with User Datagram Protocol (UDP), using code based on [5]. The distance between Ground Control Station (GCS) and the Pilot is relatively close – about 5 meters- where the distance between the UAV and the Pilot is at maximum 50 meters (the HoloLens visibility is limited, thus, there is no point to fly further than 50 meters). This requires a local wi-fi network, provided by an external router (such as a cellphone access point). Initially, two-way communication via UDP was planned: the ground station sends data about the drone and the set exercise to HL2, and HL2 sends back the data processed by the scoring algorithm. However, during testing, considerable difficulties with sending data packets from HoloLens 2 were encountered. A simplified version of it functions on HL2, while the GCS uses the complete algorithm to generate the report. Such a solution allows the use of the full functionality of flight ratings at the ground station without using the glasses. This is a major advantage since the HoloLens 2 glasses are relatively expensive. The system, which does not require a holographic device to be placed on the pilot's head, can also be more easily combined with currently existing training procedures, as no changes need to be made to them. The increased modularity of the system is also perfectly in line with the high degree of configurability assumed from the beginning.

The HoloLens 2 glasses allow the pilot to display augmented reality holograms depicting the flight route, obstacles, landmarks, position, and orientation of the drone, as well as indications of data such as air speed and instantaneous quality assessment in the form of a HUD, among others.

The system can operate in two modes: free flight and recorded flight. The latter is distinguished by the fact that the moment of its start and end is controlled by a ground station, and during its duration, all flight parameters are recorded. Based on

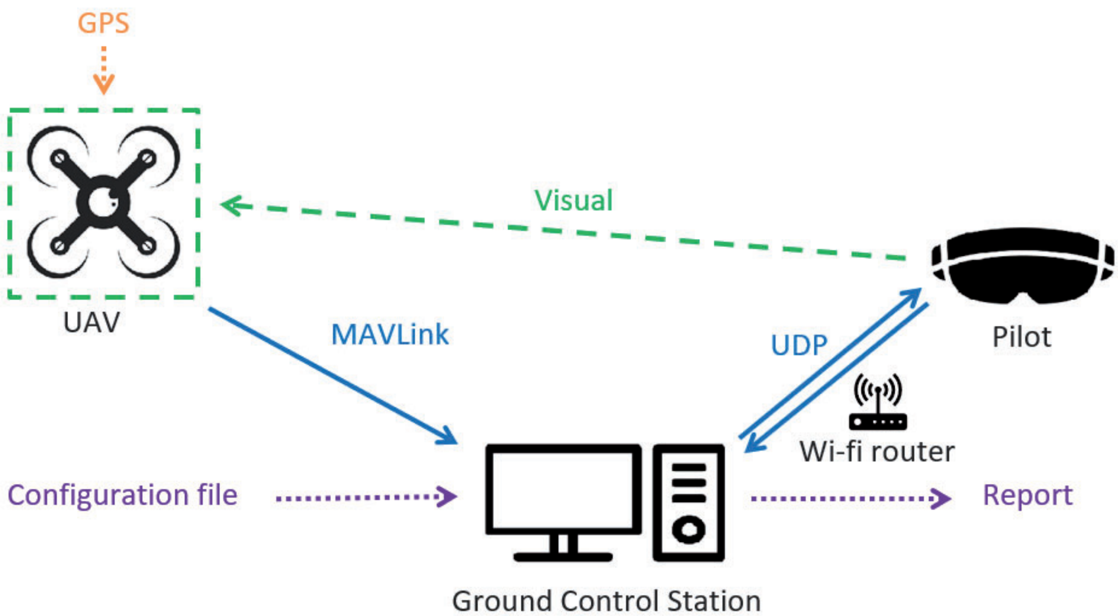


Fig. 1. System structure (User Datagram Protocol (UDP), MAVlink communication protocol, Unmanned Aerial Vehicle (UAV).

Tab. 1. Configuration file parameters.

Parameter	Value	Comment
Title & Author	Required	For identification purposes
Operator coordinates	Required	Required to calculate relative position to drone
Waypoints	Optional	3D coordinates of flight path's subsequent points, either in GPS or local coordinates
Markers	Optional	3D coordinates of white circles placed on the ground
Obstacles	Optional	3D coordinates of obstacles that should be avoided
CoordsInGPS	0 or 1	1 if Waypoints, Markers and Obstacles are given in GPS coordinates, 0 if in local HL2 coordinate system
AltitudeDiffIgnore	0 or 1	Whether the altitude of the UAV is scored by the algorithm
GuidelineOffset	0 by default	Offset in meters of the flight path visualization from the set flight path (e.g., the exercise can require flying the UAV to fly a set height above a line shown on the ground).
DistanceWeight	≥ 0	Distance error weight for the scoring algorithm
AttitudeWeight	≥ 0	Orientation error weight for the scoring algorithm
SpeedWeight	≥ 0	Speed error weight for the scoring algorithm
TargetSpeed	≥ 0	Set speed in m/s
Accuracy	≥ 0	How close the UAV has to get to a waypoint for the system to select the next waypoint as target, in meters
PausingTime	≥ 0	Set hover time at waypoints, in seconds
Target	-1; -2; [0;360)	-1 for next waypoint. -2 for none set. [0;360deg) for constant for the entire exercise
ShowWaypoints	0 or 1	1 to visualize waypoints in HL2, 0 to hide
ShowGuideline	0 or 1	1 to visualize flight path in HL2, 0 to hide
ScoringMethod	[0;1]	0 for Mean Absolute Error, 1 for Root Mean Square Error, value in between for weighted average

this, a report is generated at the end, including an overall assessment of the flight along the set route. It was decided that the flight route and other conditions of the exercise will be defined in text format in a configuration file, a necessity that would arise at the initiation of the ground station server. Also included in this file will be the operator's geographic coordinates from HL2, necessary to properly display the drone's position in the goggles. The list of all the parameters in the configuration file can be found in Tab. 1.

In order to ensure the calibration process is both reliable and precise, it is crucial for the operator to hold a specific position. However, this requirement is in line with the method of training without the developed system. Every effort has been made to make the configuration file suitable for quick and easy creation and modification by the instructor.

Position Calculation

The display system in HoloLens 2, as well as the scoring algorithm, uses a local coordinate system XYZ. In this system, positive X is positioned to the right of the HL2, with positive Y oriented upwards. The calculation of relative distances is performed in meters. Thus, one of the first key tasks in the system is converting the drone's and operator's GPS-obtained positions, given in terms of latitude ϕ and longitude λ, to the aforementioned local coordinate system. An algorithm based on the WGS84 model was used for this purpose, using formulas derived in [5]. The algorithm is available written in C# as a library for Unity [10] and, for the purposes of this project, has been rewritten in Python. In order to convert geographic coordinates to the local Cartesian coordinate system, the number of meters per degree of longitude and latitude at the designated location is first determined. The equations (1) and (2) are used for this purpose:

$$r = R_N \cos\emptyset = \frac{a}{\sqrt{1-e^2 sin^2\emptyset}} \cos\emptyset \tag{1}$$

$$R_M = \frac{a(1-e^2)}{(1-e^2 sin^2\emptyset)^{3/2}} \tag{2}$$

where:

- r is the radius of the parallel
- R_N is the radius of curvature in the prime vertical.
- R_M is the radius of curvature of the meridian.
- α is the semi-major axis of the WGS84 ellipsoid.
- e is the first eccentricity of the WGS84 ellipsoid

Knowing that a circle encompasses a solid angle of 360° and the equations for the radii r and R_M, we can determine using the formula for the circumference of the circle the following coefficients k, as a function of latitude ϕ:

$$k_\lambda(\emptyset)\left[\frac{m}{\circ}\right] = \left[\frac{\pi}{180}\right]\frac{a}{\sqrt{1-e^2 sin^2\emptyset}} cos\emptyset \tag{3}$$

$$k_\emptyset(\emptyset)\left[\frac{m}{\circ}\right] = \left[\frac{\pi}{180}\right]\frac{a(1-e^2)}{(1-e^2 sin^2\emptyset)^{3/2}} \tag{4}$$

The functions shown in the equations (3) and (4) are even and can, therefore, be approximated using the Fourier cosine series:

$$k_\lambda(\emptyset)\left[\frac{m}{\circ}\right] = p_1 cos\emptyset + p_3 cos(3\emptyset) + p_5\cos(5\emptyset) \tag{5}$$

$$k_\emptyset(\emptyset)\left[\frac{m}{\circ}\right] = m_0 + m_2\cos(2\emptyset) + m_4\cos(4\emptyset) + m_6\cos(6\emptyset) \tag{6}$$

where: p1 = 11412.84, p3 = -93.5, p5 = 0.1177 and m0 = 111132.92, m2 = -559.82, m4 = 1.175, m6 = -0.0023. With the values of the k coefficients, the horizontal position of the UAV relative to the position of the operator can be calculated.

Scoring algorithm

The application of a scoring criterion based on the value of the error necessitates the prior definition of the error itself. According to the assumptions, its value should be influenced by three factors:

- the drone's position error,
- the drone's orientation error,
- the drone's speed error.

In addition, the scoring algorithm should be user-configurable, which is ensured by adding weights to each of the error components. The error e(i) was thus defined as follows:

$$e(i) = \frac{w_d e_d(i) + w_a e_a(i) + w_v e_v(i)}{w_d + w_a + w_v} \tag{7}$$

where:

- e_d is the distance error, defined as the shortest distance between the position of the UAV and any part of the set flight path, in meters,
- e_i is the orientation error, defined as the modulus of the angle between the UAV's orientation and the set orientation (defined in the configuration file), in radians,
- e_v is the speed error, defined as the modulus of the difference between the UAV's speed and the set speed (defined in the configuration file), in meters per second,
- w_d is the distance error weight, whose value is defined in the configuration file,
- w_a is the weight of the orientation error, whose value is defined in the configuration file,
- w_v is the weight of the speed error, whose value is defined in the configuration file,
- i is the time step designation.

In [11], an algorithm to objectively assess the quality of object control has been developed. Various criteria are described and used, amongst others the Integral Absolute Error, Mean Absolute Error, Integral Square Error, Mean Square Error, Root Mean Square Error, Integral of Time-weighted Absolute Error, Integral of Time-weighted Square Error, and Integral of Square Error Divided by Time. Criteria giving errors a weight that varies with time are described as being able to consider the pilot's initial habituation to the control and his increasing fatigue over time. It was found that in the planned applications of the system designed in this work, there is no need for these solutions. The

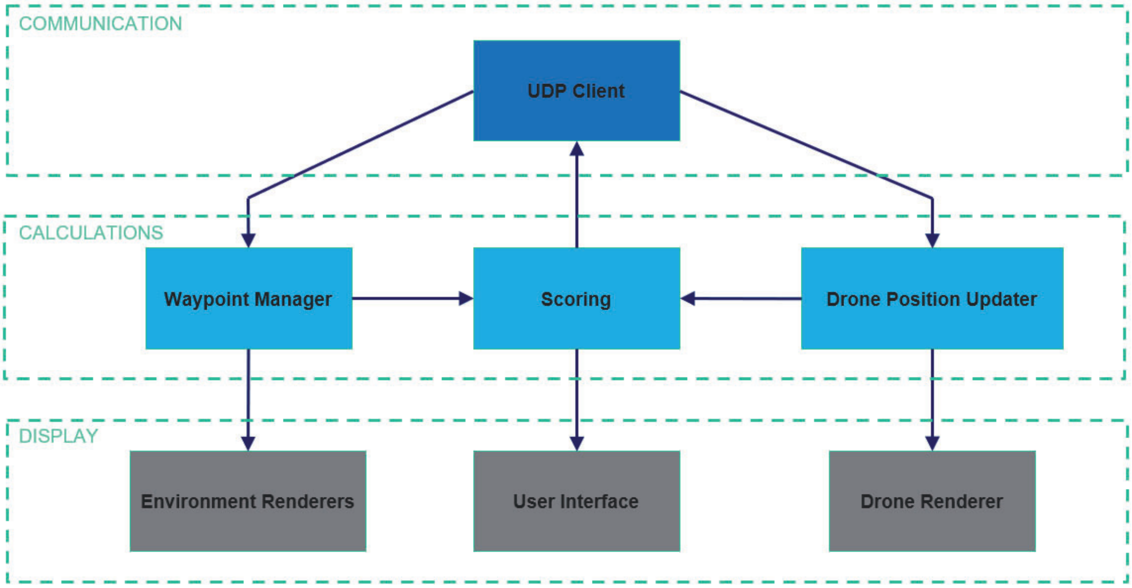


Fig. 2. Unity game engine architecture.

main reason is the duration of the exercise, which is a few minutes at most. The pilot getting used to the controls or growing fatigue will be evident in the form of better or worse results in subsequent flights, not over the course of a single exercise. Criteria operating with a continuous error value were also excluded since the telemetry data transmitted by the UAV is discrete in nature. The aforementioned criteria were used in [11] to evaluate simulated flights of UAVs performing given exercises. It was concluded that RMSE would be the best for this type of application , as was explained in [11].

The RMSE criterion meets all the assumptions and requirements set in the project regarding the evaluation algorithm, so it was selected for use in the system. RMSE, however, is characterized by giving higher weights to errors with larger values, which may or may not be desirable for the instructor-user. In order to increase the configurability of the system, it was decided to implement a second criterion that does not assign any weights, MAE, to the system. The user will have the choice of using one of the criteria or a weighted combination of both. The Root Mean Square Error criterion used in the system is shown in eq. (8), and the Mean Absolute Error criterion in eq. (9):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{i=n} e^2(i)} \tag{8}$$

$$MAE = \frac{1}{n}\sum_{i=0}^{i=n}|e(i)| \tag{9}$$

where *e(i)* is the error calculated in eq. (7), and *n* is the total number of time steps.

Thus, the score will be calculated according to the following formula:

$$S = 100(wRMSE + (1-w)MAE) \tag{10}$$

or, substituting eq. (8) and eq. (9) into eq. (10):

$$S = 100(w\sqrt{\frac{1}{n}\sum_{i=0}^{i=n} e^2(i)} + (1-w)\frac{1}{n}\sum_{i=0}^{i=n}|e(i)|) \tag{11}$$

Mixed reality application

Using the Unity game engine, an app for HoloLens 2 has been developed. Its architecture is shown in Fig. 2. In the application, the *UDPClientManager* object, shown as the *UDP Client* block in Fig. 2, is responsible for receiving data sent by the server via UDP and passing it on to the blocks responsible for the drone's positioning and environment. The Drone Position Updater is attached to the drone object in Unity, receives data from the UDP Client block, and changes the position and orientation of the drone accordingly. The Waypoint Manager block is responsible for the exercise environment displayed in the glasses. Contrary to the name of the block, the environment does not consist only of waypoints but of three optional elements:

- the assigned flight route (called in the project guideline), which is defined by waypoints and takes the form of a series of connected line segments,
- obstacles, which are cuboids that the drone should avoid,
- markers, white flat circles that are designed to be placed at ground level.

The *Waypoint Manager* processes the environmental data according to the system configuration and passes it to the application elements responsible for visualizing the environment, called *Environment Renderers*.

The Scoring block is the application element responsible for checking the drone's interaction with the environment. It calculates the error as shown in eq. (7), as well as checks for potential collisions with obstacles. If the exercise requires it, a target selection algorithm that checks the order in which the waypoints are reached is enabled. The algorithm has no chosen target at the beginning of the exercise. When the UAV reaches the first waypoint (is closer to it than the radius defined in the config file), the second waypoint is selected as the target. This process is repeated with each successive waypoint, until the last one, at which the algorithm reverts to not having the chosen target and allows flying the route again. It is permissible to skip a waypoint, while returning to an already passed waypoint will not undo the selected target. If a hovering waiting time is set at waypoints, the target will jump only after this time has elapsed.

In the HoloLens 2, the user interface is visible during flight (for the operator), in the form of parameters displayed to the pilot on the HUD. Its appearance is shown in Fig. 3, with no connection to the GCS server. The following elements can be found on it:

– the left side shows the status of the connection to the GCS server in the form of its IPv4 address (in Fig. 3, only "IP SERVER" is shown, indicating no connection),
– in the middle-upper part of the screen, the current target is displayed (waypoint number or required orientation, or nothing if not set),
– the right side displays flight parameters, such as the drone's speed, its distance from the guideline or deviation from the preset orientation,
– under the target information in the upper center of the screen, the UAV's yaw is shown, and the pitch and roll can be seen on the right,
– at the bottom of the screen, the current error value is given in large font, labeled as Score,
– at the very bottom, the position of the drone in the local EUN system is given, which can be useful when calibrating the system,
– at the center of the screen a warning can be seen. It appears if and when a collision with an obstacle occurs.

Fig. 3. The User Interface displayed in HoloLens 2 as seen by the operator.

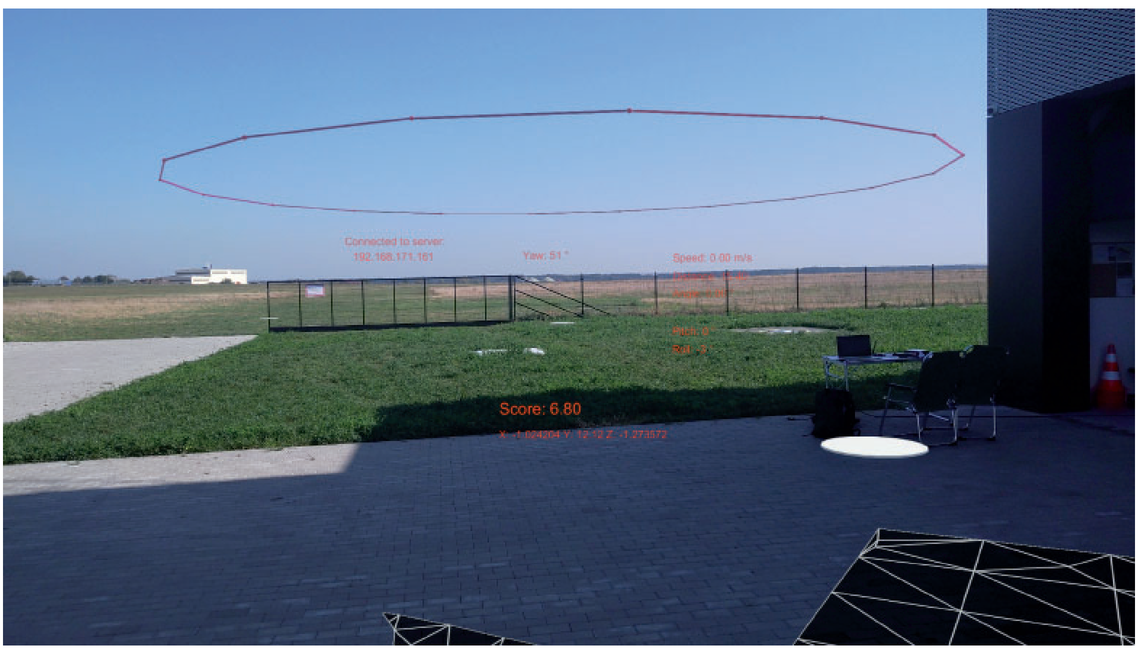


Fig. 4. Visualization of the exercise E1 in the HoloLens 2 app.

Fig. 5. Visualization of exercise E2 in the HoloLens 2 app.

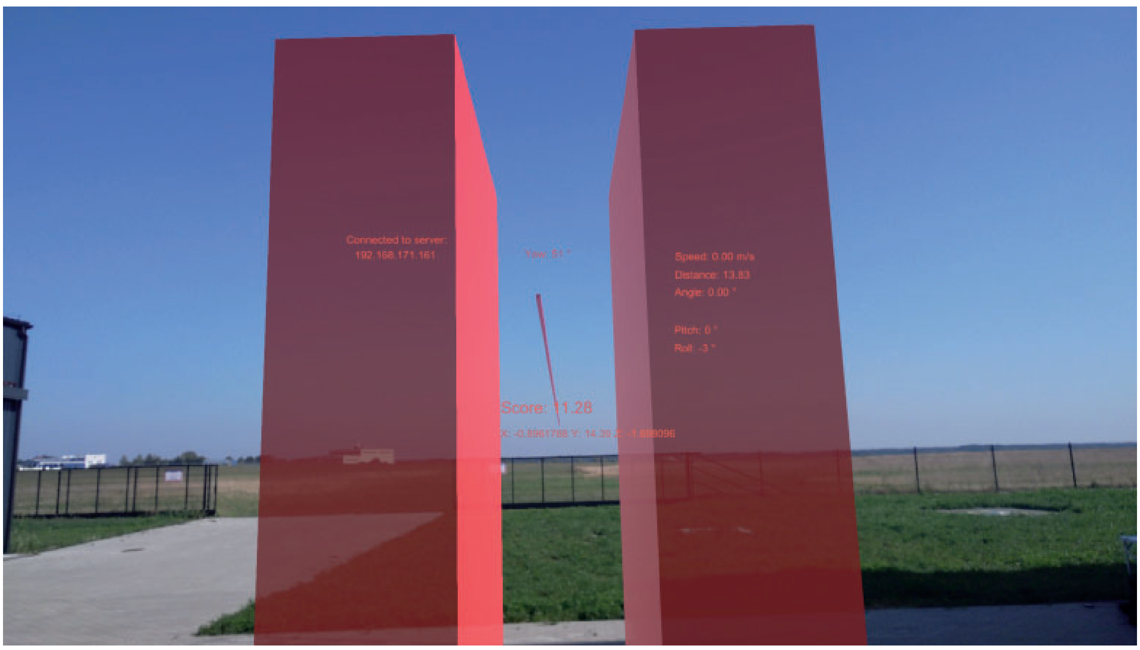


Fig. 6. Visualization of the third exercise in the HoloLens 2 app.

Exercises

In 2019, the Polish Civil Aviation Authority issued documents [13] describing exercises required for a UAV operator to be certified for visual line of sight flights. Based on them, as well as on [12], configuration files for exercises were written, with the intention of testing and demonstrating the functionality and capabilities of the system. For the first tests, a set of three tasks were analyzed:

Exercise 1: Horizontal circling

This exercise was developed based on exercise 5 in [12], which involves flying in a circular pattern while maintaining the UAV's forward orientation and constant altitude (Fig. 4). The scoring algorithm weights used are $w_v = 0$ and $w_d = w_a = 1$. The target selection algorithm is enabled, with the UAV required orientation towards the next waypoint.

Exercise 2: Horizontal Eight

This exercise was developed based on exercise 6 in [11], which involves flying a figure-eight, maintaining UAV's forward orientation and constant altitude (Fig 5). The UAV is required to fly along a horizontal figure-eight consisting of two circles ten meters in diameter each. The weights used are $w_v = 0$ and $w_d = w_a = 1$. The target selection algorithm is enabled, with the UAV required orientation towards the next waypoint. It is necessary to fly the route twice without stopping. It is performed by starting from the center of the figure eight in a counterclockwise direction, covering the entire figure eight twice.

Exercise 3: Flying through a corridor between obstacles

This exercise was developed based on the additional exercise E1 from [7], which involves flying between two obstacles forming a narrow corridor. A straight-line flight path between two obstacles is set. The goal of this exercise is to avoid collision with obstacles during the flight. As such, the numerical score calculated by the system is not to be paid attention to. In case of a collision, a red caption appears in the report: *Collisions: YES* – which means the exercise has been unsuccessful.

Airfield tests

To check the system components' performance on real hardware, tests on the ground were performed. The quality of communication, the delays occurring in the system, and the accuracy of position determination were checked.

The drones used for the tests were equipped with a Here 3 GPS module, which, without Real-Time Kinematic positioning (RTK), has a positioning accuracy of 2.5 m [9] at full satellite availability. A position determination accuracy test was conducted. First, the operator's coordinates were determined using Google Maps. Then the drone was placed stationary on the ground and the system was turned on, with the flight path set as a circle with a radius of 4 m for comparison.

Flight tests

To evaluate the system's performance in real flight conditions, flight tests were performed. The operation of the designed system was tested while flying a real UAV, a quadrotor aircraft with ArduPilot software on a Pixhawk hardware platform. It was equipped with a Here 3 GPS (CubePilot, USA) module without RTK and transmitted telemetry via an RFD 900x radio modem. The initial tests, however, were aimed to verify if the system, architecture, and communication work properly. The best mitigation of this problem is to repeat the tests with the UAV equipped with the RTK system.

RESULTS

Simulations

Simulated flights were conducted for three different exercises. There were two maneuvers taken – circle and horizontal eight and obstacle avoidance. The results are presented in similar patterns, showing the trajectory of the drone (desired and realized) followed by detailed data driven from the telemetry of the drone. All those data were subject to the algorithm (Eq. 9) and the final result was calculated.

Exercise 1: Horizontal circling

The visualization of the exercise in HoloLens 2 is shown in Fig. 4. The simulated UAV was flown along a horizontal circle with a diameter of 30 m. The set flight path can be seen in the sky. The spot for the operator to stand in during the exercise is marked by a white circle (marker). In the lower right corner, a few artefacts of the HL2 inbuilt object detection are visible, but they have no effect on the workings of the designed system. The results obtained are shown in figures 7, 8 and 9. Figure 7 presents the trajectory of the flight. In this case, the altitude was held automatically (seen in Figure 8 – yellow plot). The flight was scored by the algorithm as 0.72 using RMSE and 0.52 using MAE. It is worth paying special attention to the yaw graph (Fig. 8), where it can be seen that the target orientation varies in an approximately linear fashion as a result of the target selection algo-

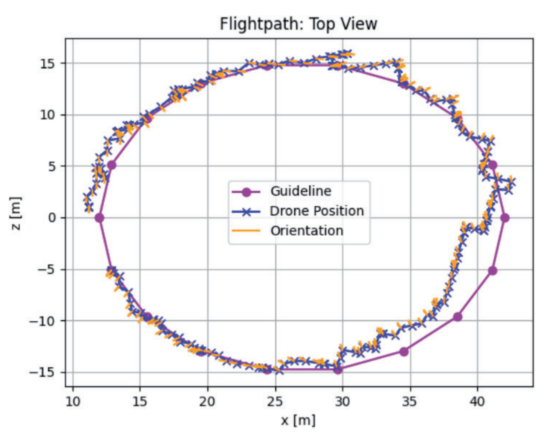


Fig. 7. Drone flight trajectory in the simulated test flight for E1.

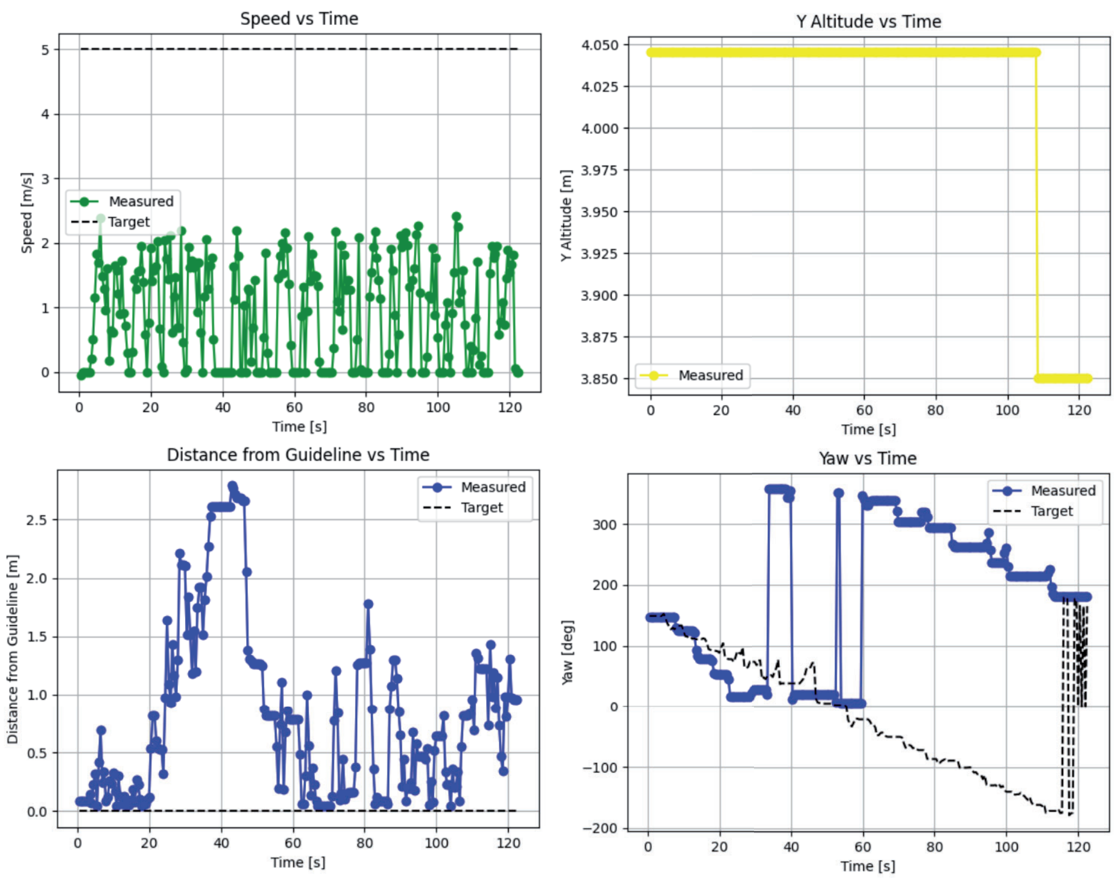


Fig. 8. Drone flight parameters in the simulated test flight for exercise E1.

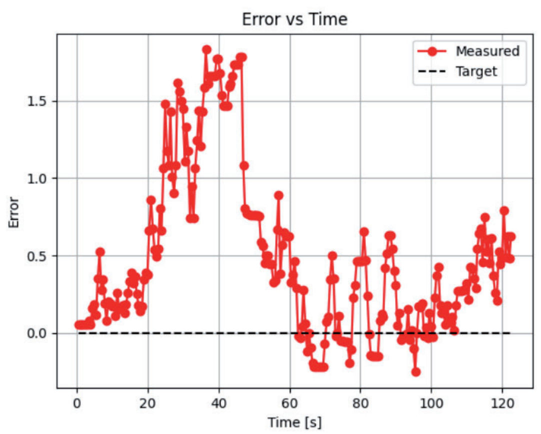


Fig. 9. The value of the error calculated by the system (as described in eq. (10)) in the simulated test flight for exercise E1.

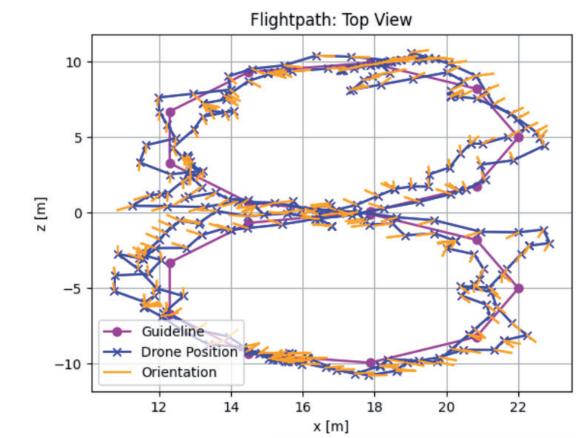


Fig. 10. Drone flight trajectory in the simulated test flight for exercise E2.

rithm working during the flight. The data received were subjected to an assessment algorithm (Eq. 11), so the results were generated. In the Figure 9, the error calculated from RMSE is presented. The graph provides a visual representation of the operator's error rate over time. In this particular case the most significant factor was the distance (in meters) from the original path after travelling half of the circle. The wages of the error may be subject to modification, given the criteria that were applied to the task in question. The subsequent waypoints targeted by the system for orientation scoring are shown as dots on the set flight path (guideline).

Exercise 2: Horizontal Eight

The visualization of the exercise in HoloLens 2 is shown in Fig. 10. The results obtained are shown

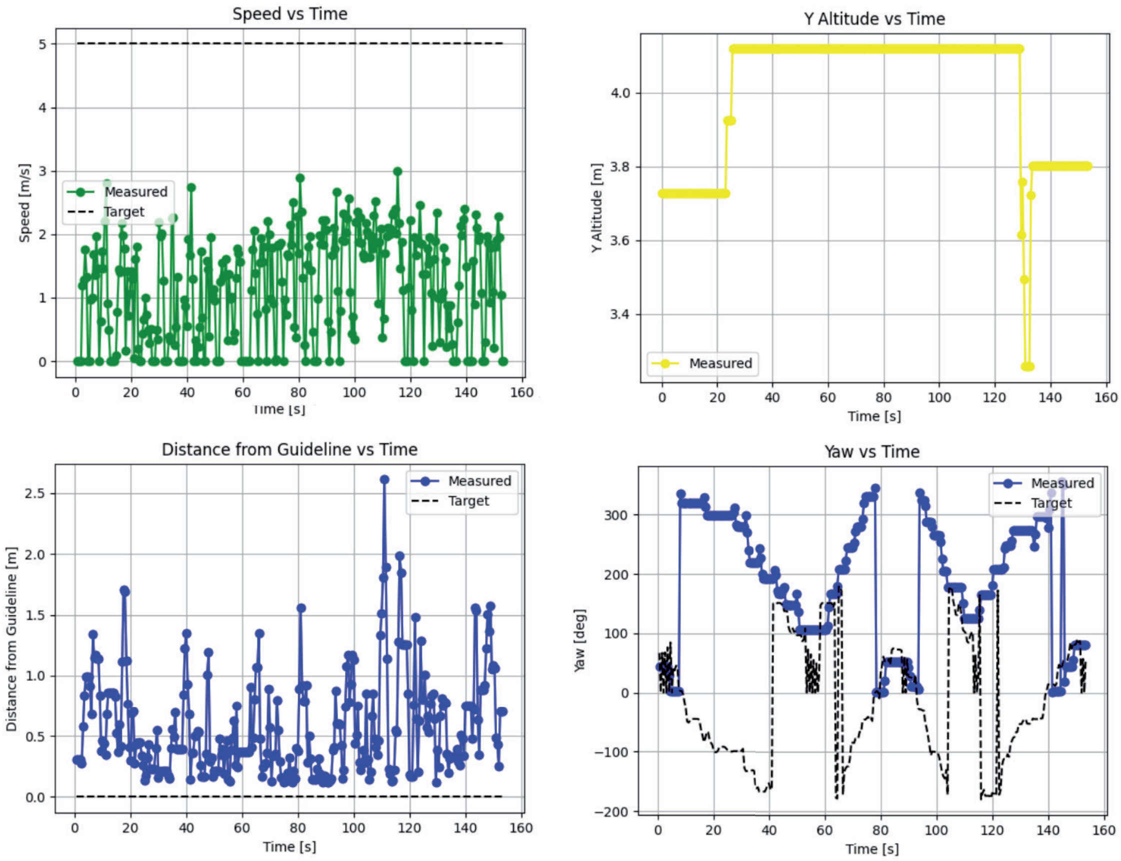


Fig. 11. Drone flight parameters in the simulated test flight for exercise E2.

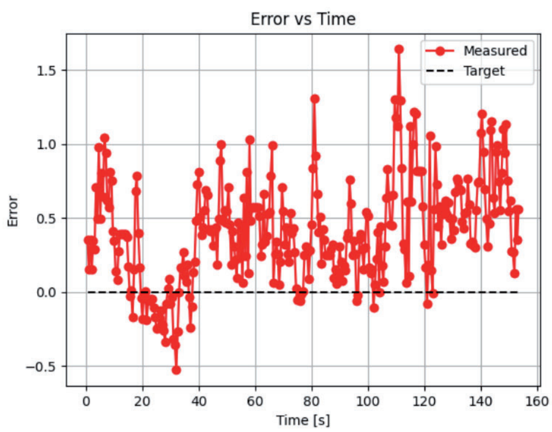


Fig. 12. The value of the error calculated by the system in the simulated test flight for exercise E2.

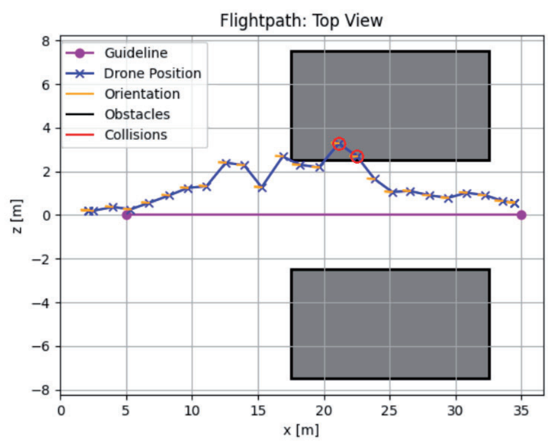


Fig. 13. UAV flight trajectory in the simulated test flight for exercise E3. Collisions with the virtual obstacle can be seen marked with a red circle.

in Figs 9, 10, and 11. The flight was scored by the algorithm as 0.54 using RMSE and 0.44 using MAE. It was performed, as required by the exercise, by starting from the center of the figure eight in a counterclockwise direction along the upper (Fig. 9) circle.

Exercise 3: Flying through a corridor between obstacles

The visualization of the exercise in HoloLens 2 is shown in Fig. 6. A straight-line flight between two obstacles was performed. No significant maneuvers are performed, and the numerical score is ignored for this exercise, so only the flight path graph is shown in Fig. 13. It is important to acknowledge that a short collision may not always be recorded, particularly in instances where there is a gap between successive position indications by the UAV. Such a situation could have happened here as shown in Fig. 13, where it can be seen that the flight path slightly touches the corner of the obstacle before the following two collision indications. Thus, the result is presented in the report showing the place of collision.

Airfield tests

During communication tests on real hardware, it was discovered that sending data through UDP from the glasses to the computer, which works perfectly in Unity play mode, does not work on the actual HoloLens 2 device. The cause of the problem is most likely the incompatibility of the libraries used for communication over UDP with HL2 glasses. If this is indeed the case, fixing it would require using a unique way of sending data. The problem was bypassed by duplicating the evaluation algorithm, as was already mentioned in the system design section. Communication in the other direction, i.e., sending data from the computer to HL2, works seamlessly. No objective way has been found to measure the time between sending data from the GCS and displaying the corresponding information on the glasses, but it was estimated by the operators to be under one hundred milliseconds, which is a perfectly satisfactory speed. No reduction in transmission speed or errors was noticed even when increasing the amount of transmitted data to the limits of what can sensibly be used in the exercise, considering the visibility of holograms in HL2 (i.e., a few dozen waypoints, obstacles, and markers).

Somewhat worse is the speed of the transmission of telemetry data from the UAV to the computer. The GCS's measurable data reception time averages about 0,5 seconds, while the noticeable delay in visualizing the drone relative to the UAV's actual movement is between 1 and 3 seconds, depending on the frequency of telemetry transmission. The only possible solution here is to maximize the parameter responsible for the frequency of telemetry transmission in ArduPilot, the maximum value of which is 8 Hz. The effect of the delay, however, is that the pilot cannot rely too much on the visualization of the drone in the system, even for training purposes.

The results of the position determination accuracy test are shown in figures 14 and 15. The actual position of the drone was in the EUN system: (x, y, z) = (0.5; 0.2; 0). As can be seen in the figures, in this trial, the obtained position determination accuracy is within the assumed 2.5 meters. It can also be seen that the GPS-indicated height of the UAV increased over time, starting from zero and increasing by about 5 cm per minute. Given that the exercise should take a maximum of a few minutes to complete and that the GPS position accuracy is 2.5 m, the impact of the altitude indication increasing in this way is zero. This test was conducted multiple times with comparable results. The accuracy of the GPS position determination obtained in the trials is fully satisfactory. Accuracy at the level of 2.5 m makes it possible to perform and evaluate the set exercises, with the only condition being that if enabling the target selection algorithm is required, they must be more than twice the accuracy, which is 5 [m], apart. Trials show that it is possible to use GPS devices without RTK in the system.

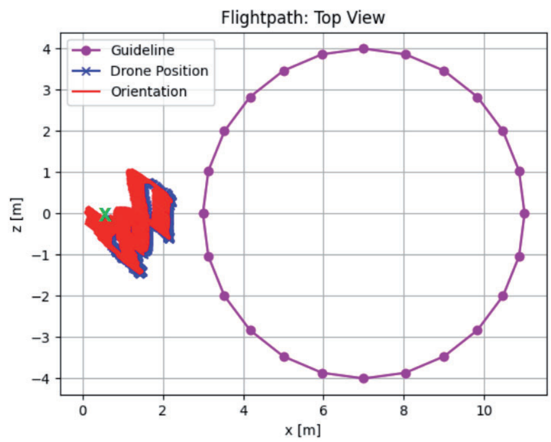


Fig. 14. Drone position measured by the system using GPS versus the real position marked with X.

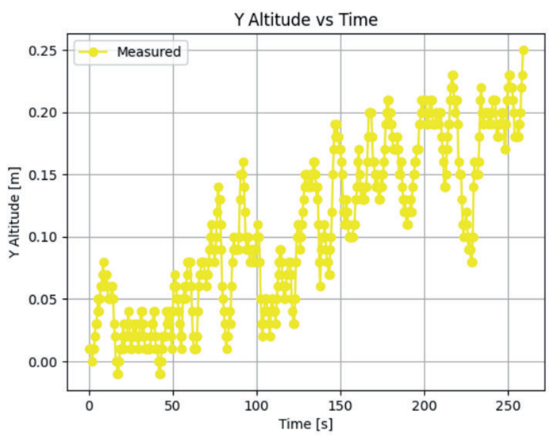


Fig. 15. Drone altitude measured by the system. The real altitude was 0.20 m.

As illustrated in Figure 16, a screenshot from HoloLens 2 presents a visual representation of an exemplary flight path, which could be utilised for an exercise. The fact that this is a screenshot is es-

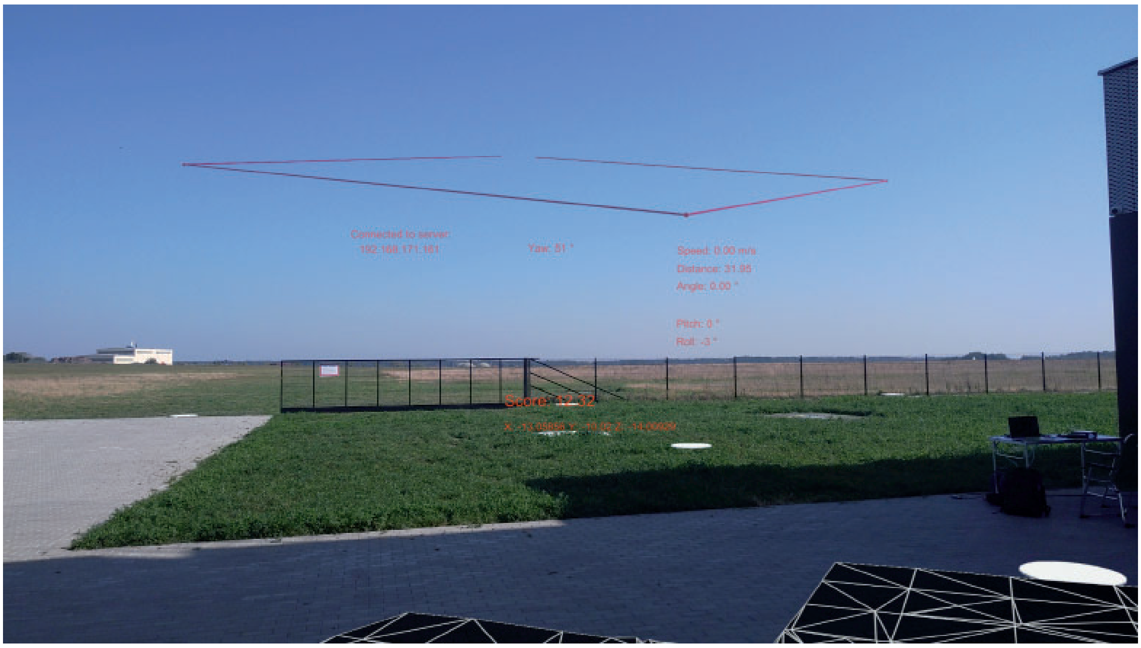


Fig. 16. Visualization of an exercise in the HoloLens 2 app, with the furthest corner of the guideline disappearing due to distance.

pecially important here because this does not represent quite exactly what the pilot sees. The User Interface is significantly bigger and more legible in reality than it is in the screenshots. There are, however, two additional factors:

– the limited field of view of the holograms, especially to the sides. Holograms that are outside the side boundaries of the HUD are very blurry in the pilot's eyes. Vertically, the visibility is better, and screenshots show upward and downward visibility relatively well,
– sunlight and overall ambient brightness. Tests were performed around noon on a day with no cloud cover. Even with the best possible display settings of the HL2, the holograms were still visible, but with less contrast than in the photos, and one had to strain one's eyes to see the more distant ones, especially at more than about 30 m distance, where they begin to blend in with the surroundings.

However, in general terms, the screenshots from HL2 demonstrate what the pilot observes. It is evident from these screenshots that the holograms are clearly three-dimensional, and there are no issues in determining their position in space. In addition, the position of the holograms is stable and stationary, they can be walked around and viewed from all sides. It has already been mentioned that from a distance of about 30 m, the holograms begin to merge with the surroundings in the pilot's eyes. At a distance of about 42 [m], they disappear completely, including in the screenshots from HL2. This can be seen in Fig. 16, in which the farthest corner of the flight path is at a distance of slightly over 42 meters from the pilot and is invisible. This imposes a sharp limitation on the exercises that can be visualized by the system.

Flight tests

During the flight tests, the telemetry transmission rate of the UAV was set at 1 Hz. This value was too low for the system's needs, and it turned out to be impossible to change. This resulted in a delay of about 3 seconds in the visualization of the drone compared to reality. The operator's position was determined using the Google Maps application and confirmed with a Here 3 GPS module. However, calibrating the system proved to be quite a problem when, despite numerous tests, during calibration flights, and modifying the operator's position, it was not possible to set the drone's visualization with greater accuracy than about 3 to 4 meters from the actual UAV. The tests took place in an open field at the Warsaw University of Technology airport in Przasnysz, so access to GPS satellites should not have been a problem. However, this challenge has to be overcome with different solutions. Despite problems with calibrating the system, a number of test flights were made, three of which are shown in this paper, as in their course the pilot flew manually, trying to follow the indicated flight path. In addition, they were performed one after the other with brief time intervals between them (a few minutes). The tasked exercise was Exercise 4: Flight around the perimeter of a square at a constant altitude. This

Tab. 2. Flight tests.

Flight No	Score (for w=1)	RSME	MAE	Flight time [s]
1	589	5.89	5.23	92.71
2	472	4.72	4.34	78.21
3	354	3.54	3.37	61.23

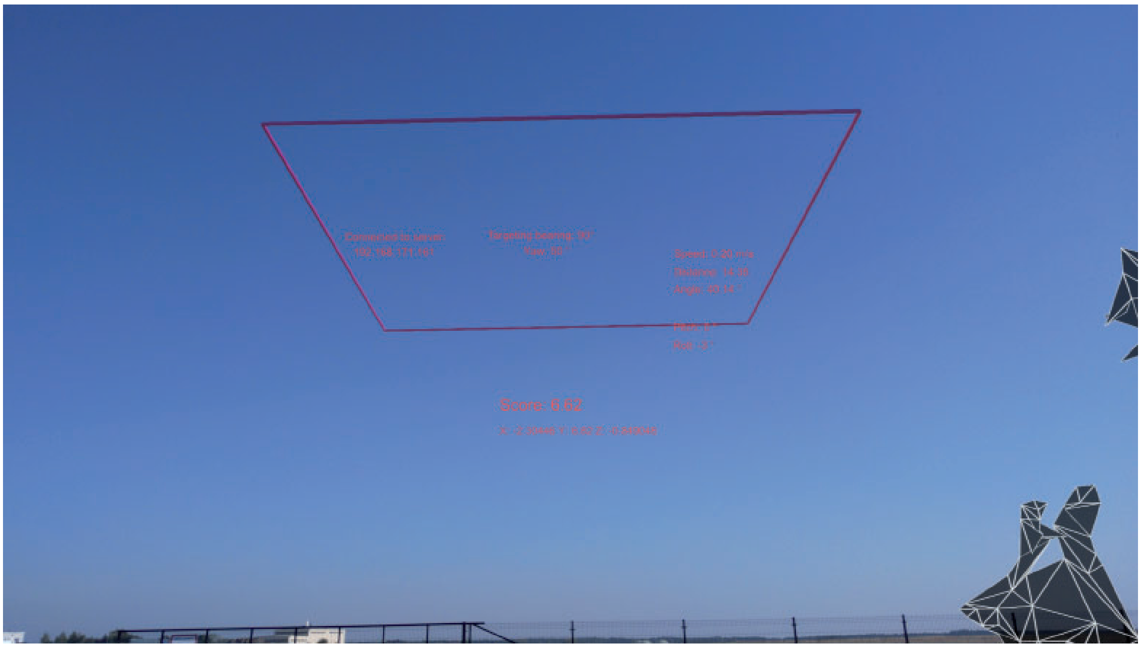


Fig. 17. Visualization of the fourth exercise in the HoloLens 2 app.

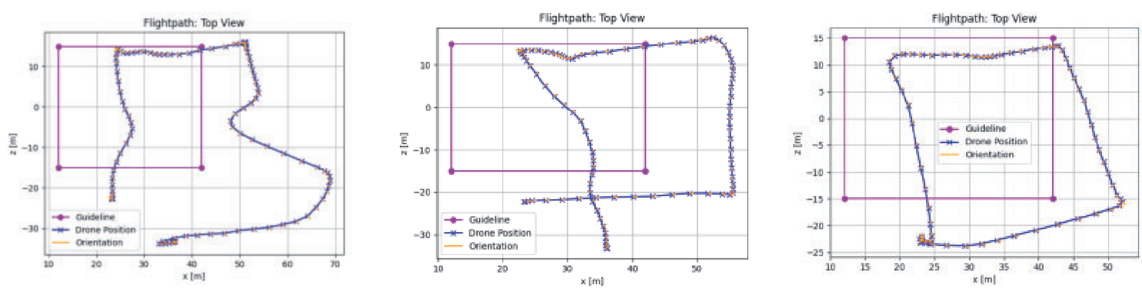
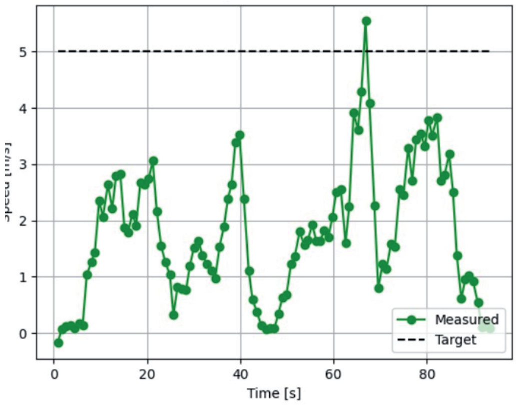


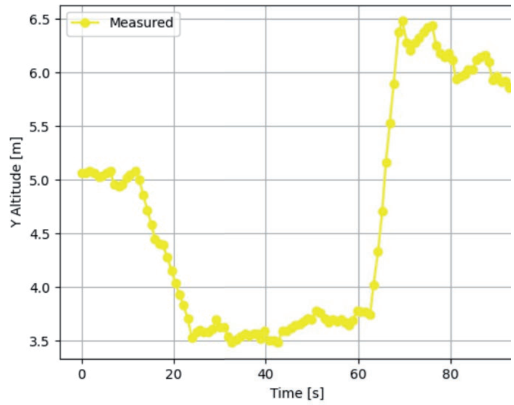
Fig. 18. Drone flight trajectory in the three test flights.

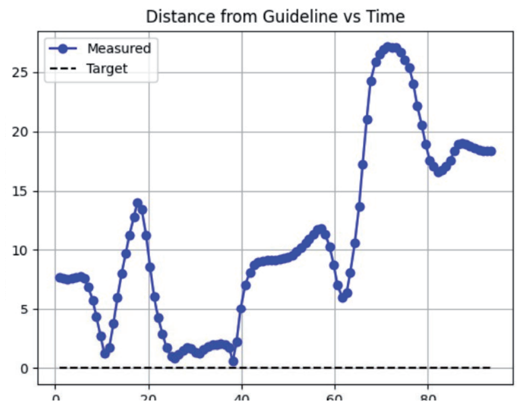
exercise was developed based on exercise 3 in [8], which involved flying in a straight line in different orientations relative to the operator while maintaining a constant altitude manually. The guideline was a square of 30 m on the side, as can be seen in Fig. 17. The tasked flight speed was 5 [m/s], orientation in the positive direction OX (i.e., east), without stopping at waypoints. The results of the flights were collected in table 1. In it, it can be seen that with each successive flight, the pilot performed better and better with the set route, flying it faster and with better results. The most likely reason for this is that the pilot was getting used to the system and, most importantly, to HoloLens 2.

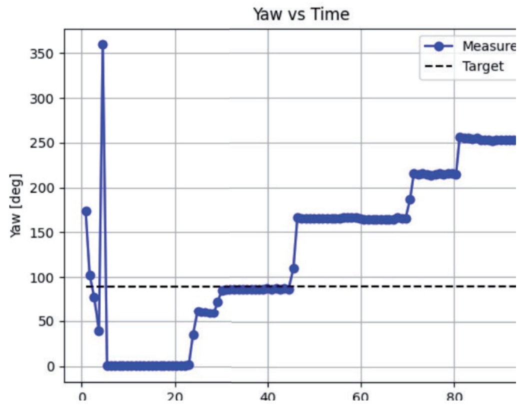
Figure 18 shows the flight trajectories in the three consecutive trials, respectively. The pilot was at the (0,0,0) point, while the sun was shining from the south, i.e., from the bottom side of the graphs. This caused the far-right corner of the route (bottom right in the graphs) to be essentially invisible to the pilot. Figure 19 shows the flight speed, altitude, yaw, and distance, while Fig. 20 shows the distance error during testing.

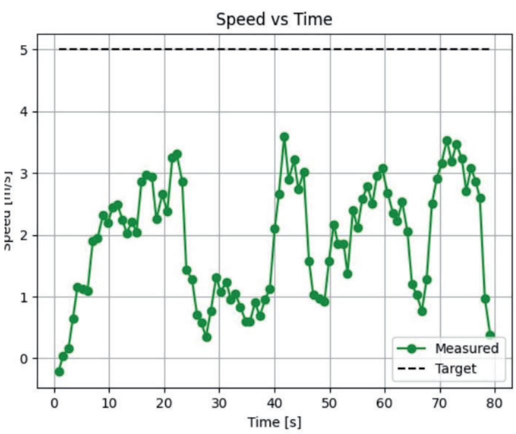
While the graphs clearly show improvement in flight quality between trials and increasingly close following of the set route, problems are also visible. The most pronounced of these is that even for the best flight, the average distance error is more than 5 [m], at times reaching almost 15 m. The second noticeable relationship is that the pilot kept the UAV closer to the set route when flying along sides parallel to the OX axis, while along

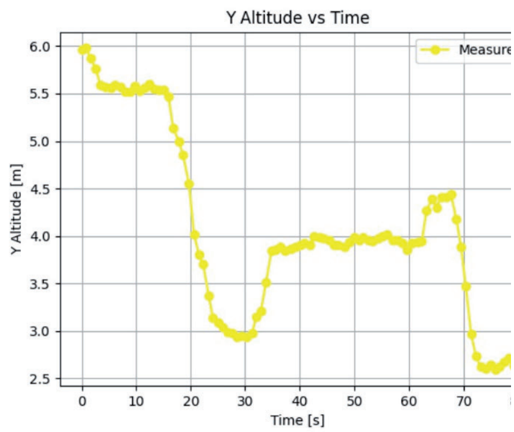

Measured
Target
Time [s]

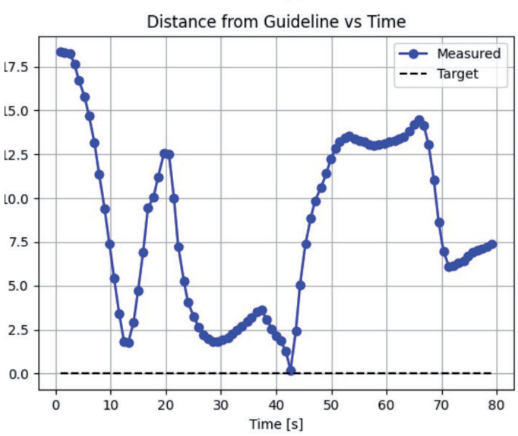

Measured
Y Altitude [m]
Time [s]

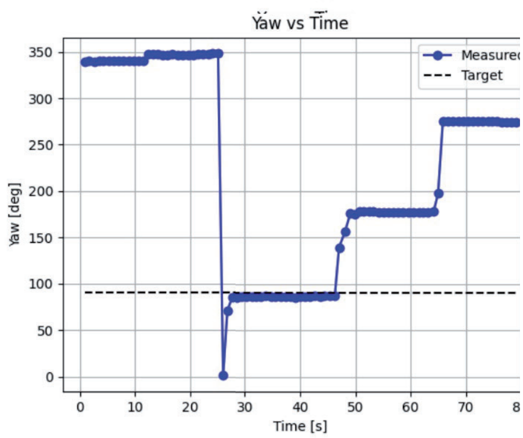

Distance from Guideline vs Time
Measured
Target


Yaw vs Time
Target
Yaw [deg]


Speed vs Time
Measured
Target
Time [s]


Y Altitude vs Time
Y Altitude [m]
Time [s]


Distance from Guideline vs Time
Target
Time [s]


Yaw vs Time
Target
Yaw [deg]
Time [s]

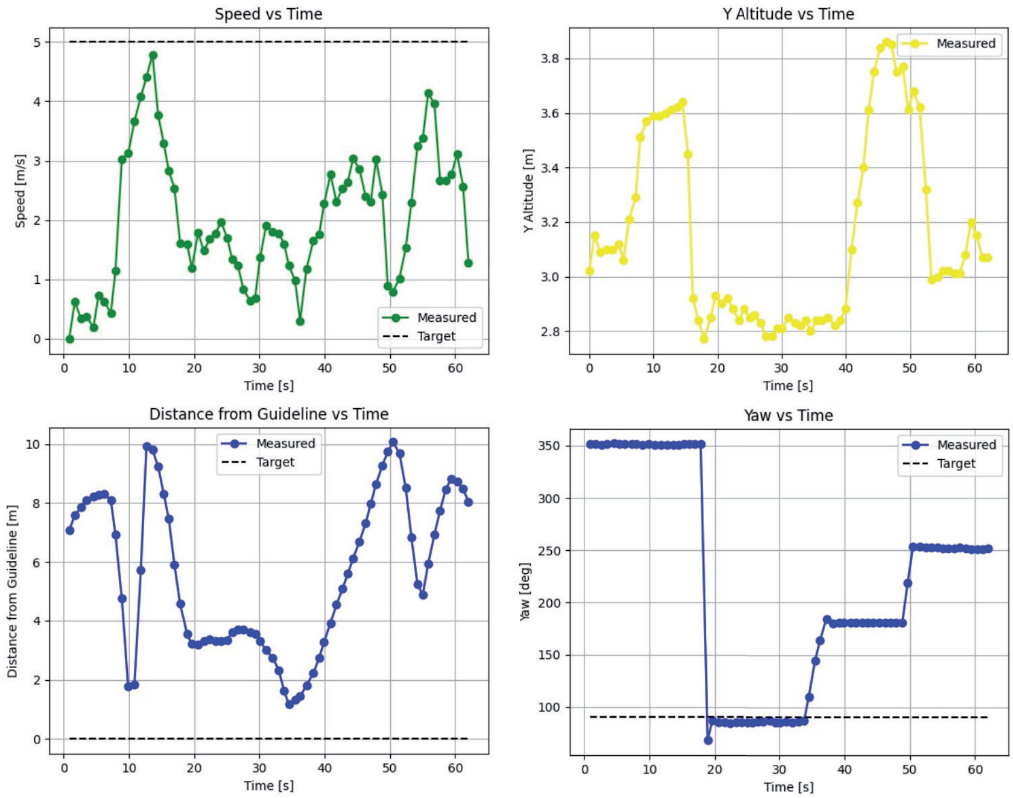


Fig. 19. Drone flight parameters (Speed vs Time, Altitude vs Time, Yaw vs Time, Distance for Guideline vs Time) in the three test flights with the real UAV.

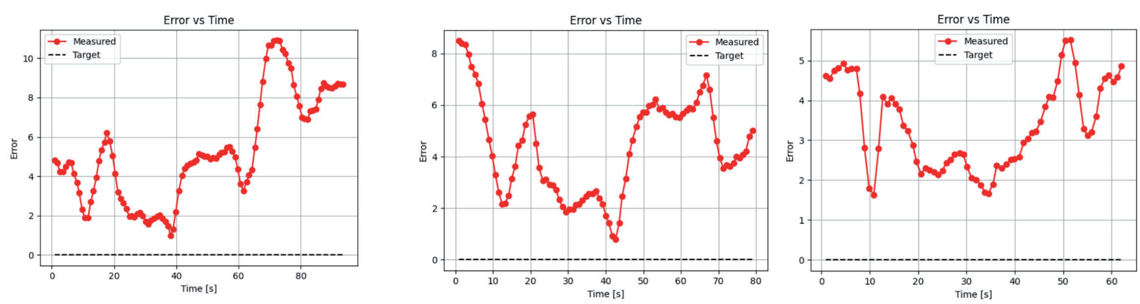


Fig. 20. The value of the error calculated by the system in the three test flights with the real UAV.

the other pair of sides, the distance could be even more than half the length of the side of the square. There is a delay in drone visualization due to the low frequency of telemetry data transmission.

- poor visibility of the holograms due to distance and sunlight,
- relatively poor accuracy of the system's position determination due to calibration issues,
- for the pilot, this was the first contact with the system in flight,
- potential difficulty for the pilot to sense the drone's position relative to the position of the guideline in the air - this would be evidenced by the dependence of the distance error value on the direction of flight, as well as quite a large fluctuation of about 3 [m] in flight altitude in the first two attempts, while the route was horizontal.

In Fig. 19, flight speed graphs (green plots) indicate an average air speed of about 2 [m/s], while the orientation graphs show that the pilot kept the drone facing the direction of flight. These values are due to the pilot's choice to fly that way rather than anything related to the designed system.

DISCUSSION AND CONCLUSIONS

This paper presented the design of a drone pilot training system using the HoloLens 2 mixed reality goggles. It was intended to solve the problem of the lack of reproducibility and subjectivity of examinations for unmanned aircraft pilot certificates. The following tests are designed to demonstrate the efficacy of the system in its entirety, including the processes of data gathering, transfer, and result calculation. The presented system may be used by both operators and instructors for the purpose of enhancing the efficiency of training. The system is also equipped with an additional data report, the analysis of which could facilitate a more detailed examination of the manoeuvres. The following assumptions were made about the designed system - it should:

- display real-time information to the pilot about the exercise, errors made, and flight parameters of the UAV. For this purpose, the system was to use HL2. From the flight test, it occurred that the delay time varying from 1-3 seconds may be a significant factor regarding the time response and the UAV reaction to malfunction. This challenge may be improved using more advanced systems for communication,
- generate an objective evaluation and score after the exercise to allow comparison between successive flights by the same pilot or between different pilots and generate an exercise report that allows the instructor to have more insight,
- fulfill the role of supporting the instructor, not replacing them,
- the system should collaborate with drones based on the popular ArduPilot autopilot software.

After completing the work on the project, it can be said that the listed assumptions were met in the project. A system consisting of two main elements has been created:

- ground station software, which, when run on a computer, provides communication between the UAV and HL2, has a scoring algorithm implemented, and allows reports to be generated,
- a HoloLens 2 app, which allows the tasked exercises to visualize the pilot and to show them information about the flight parameters of the UAV.

Both components of the system are functional and were evaluated first in simulations and then in in-flight tests at the Przasnysz Airport. It was found that the system has the potential to be incredibly useful but has some significant shortcomings. In the current form of the project, there are a few problematic issues:

- the limitations of the HoloLens 2's hologram display system, which has difficulty with bright sunlight and open spaces and does not visualize holograms beyond a certain distance. Looking from the perspective of this work, the solution would seem to be that all holograms, including the environment and drone visualizations, be displayed relatively close to the pilot. Implementing this in the present project would, however, require a different approach in setting up the exercises,
- difficulties with calibrating the drone's position displayed in the system. As it stands, the reference point for the local coordinate system that the system uses internally is the position and orientation of the HL2 glasses when the application is launched. After testing, it can be concluded that such a method is insufficiently accurate and too variable for such applications. Probably the best solution would be to use an automatic calibration system, for example, using the HoloLens 2 cameras to determine the initial position of a drone marked in some way. Such a solution would also allow calibration corrections to be made while the system is operating, possibly using some form of data integration,
- from the tests on the airfield, it turned out that the UAV's GPS position accuracy is quite poor. Additionally, the frequency of telemetry transmission was found to be impossible to set above 1 Hz in the tested ArduPilot configuration. Even with the expected maximum of 8 Hz, this will likely be too low for the needs of the system. However, the solution for this is under development. At the WUT there is a system being built that would be an independent navigational cube from the UAV's system. The system is equipped with RTK corrections, so the GPS accuracy will be better than in the case studied in the article. The telemetry transmission rate will be improved as well. Moreover, the navigation module would be suitable for most of the UAVs – and the results would be comparable on various platforms regardless of their internal hardware.

The relatively significant delay time is a key factor that complicates the system's use online, particularly with regard to error calculation. However, the system can still be utilized by instructors or operators to observe the trajectory and final outcome of a task. The utilization of sophisticated communication systems is likely to reduce this delay time, a factor that should be thoroughly evaluated during the initial stages of development. The obtained results are encouraging, but further research is necessary to ensure the system's full functionality in UAV flight training and certification exams.

AUTHORS' DECLARATION:

Study Design: Antoni Kopyt, Cyprian Felenczak. **Data Collection:** Antoni Kopyt, Cyprian Felenczak. **Statistical analysis:** Antoni Kopyt, Cyprian Felenczak. **Graphical representation of results:** Antoni Kopyt, Cyprian Felenczak. **Manuskrypt preparation:** Antoni Kopyt, Cyprian Felenczak. The Authors declare that there is no conflict of interest.

REFERENCES

1. Biernacki M. Simulator sickness and its relationship to oculomotor activity – comparisons of a fixed-base vs. full-motion simulator. Pol J Aviat Med Bioeng Psychol. 2020;26(3):12-17. DOI: 10.13174/pjambp.24.04.2023.02.
2. Bogus S. Drone positioning. GitHub repository. Available from: https://github.com/szymek1/DronePositioning.
3. Ficco M., Palmiero R., Massimiliano R., Granata D., MAVLink Protocol for Unmanned Aerial Vehicle: Vulnerabilities Analysis, 2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 12-15 September 2022.
4. He H, Liao X. SkyroadAR: An augmented reality system for UAVs low-altitude public air route visualization. Drones. 2023;7(9):587. Available from: https://doi.org/10.3390/drones7090587.
5. Here 3 manuals. Available from: https://docs.cubepilot.org/user-guides/here-3/here-3-manuals.
6. Matt J., Flanagan H., Chao H., Evaluation and Analysis of ArduPilot Automatic Tuning Algorithm for the Roll Tracking Controller of a Small UAS, AIAA Scitech 2021 Forum, January 2021, DOI: 10.2514/6.2021-0016
7. Rosa J, Cavalheiro G, Oliveira E, Menegon J, Mendonça F. Web AR solution for UAV pilot training and usability testing. Sensors. 2023;23(18):15478. Available from: https://www.mdpi.com/1424-8220/23/18/15478.
8. Skarżyńska J., Kobos Z., Wochyński Z. Analysis of the human factor in air accidents in polish civil aviation in the years 2010-2015, The Polish Journal of Aviation Medicine, Bioengineering and Psychology, 2018, Volume 24, Issue 2
9. Snyder J. Map projections–a working manual. U.S. Geological Survey Professional Paper 1395. 1987. Available from: https://pubs.usgs.gov/pp/1395/report.pdf.
10. Taylor M. GPS encoder. GitHub repository. Available from: https://github.com/MichaelTaylor3D/UnityGPSConverter.
11. Tomaszewska PA. Opracowanie oraz implementacja algorytmu umożliwiającego obiektywną ocenę jakości sterowania obiektem uwzględniając komfort pasażera. BSc. Eng. Thesis, Faculty of Power and Aeronautical Engineering, Warsaw University of Technology; 2019.
12. Załącznik nr 2 do ogłoszenia nr 7 Prezesa Urzędu Lotnictwa Cywilnego z dnia 6 lutego 2019 r. Urząd Lotnictwa Cywilnego - program szkolenia UAVO VLOS. 2019. Available from: https://edziennik.ulc.gov.pl/DU_ULC/2019/12/Zalacznik2.pdf.
13. Zebrowski K. System do automatycznej oceny jakości sterowania bezzałogowym statkiem powietrznym wykorzystując wirtualne środowisko. BSc. Eng. Thesis, Faculty of Power and Aeronautical Engineering, Warsaw University of Technology, 2021.