Automated, Objective Assessment of Pilot Performance in Simulated Environment

Maciej Zasuwa, Grzegorz Ptasinski, Antoni Kopyt

***Abstract*—**Nowadays flight simulators offer tremendous possibilities for safe and cost-effective pilot training, by utilization of powerful, computational tools. Due to technology outpacing methodology, vast majority of training related work is done by human instructors. It makes assessment not efficient, and vulnerable to instructors' subjectivity. The research presents an Objective Assessment Tool (gOAT) developed at the Warsaw University of Technology, and tested on SW-4 helicopter flight simulator. The tool uses database of the predefined manoeuvres, defined and integrated to the virtual environment. These were implemented, basing on Aeronautical Design Standard Performance Specification Handling Qualities Requirements for Military Rotorcraft (ADS-33), with predefined Mission-Task-Elements (MTEs). The core element of the gOAT enhanced algorithm that provides instructor a new set of information. In details, a set of objective flight parameters fused with report about psychophysical state of the pilot. While the pilot performs the task, the gOAT system automatically calculates performance using the embedded algorithms, data registered by the simulator software (position, orientation, velocity, etc.), as well as measurements of physiological changes of pilot's psychophysiological state (temperature, sweating, heart rate). Complete set of measurements is presented on-line to instructor's station and shown in dedicated graphical interface. The presented tool is based on open source solutions, and flexible for editing. Additional manoeuvres can be easily added using guide developed by authors, and MTEs can be changed by instructor even during an exercise. Algorithm and measurements used allow not only to implement basic stress level measurements, but also to reduce instructor's workload significantly. Tool developed can be used for training purpose, as well as periodical checks of the aircrew. Flexibility and ease of modifications allow the further development to be wide ranged, and the tool to be customized. Depending on simulation purpose, gOAT can be adjusted to support simulator of aircraft, helicopter, or unmanned aerial vehicle (UAV).

***Keywords*—**Automated assessment, flight simulator, human factors, pilot training.

I. Introduction

THE virtual environment offers tremendous, unique possibilities in terms of civilian and military pilot training process. Future pilots are placed in the safe environment, which prepares them for efficiently carrying out the flight procedures. Moreover, training in a simulator, serves as a very useful tool for practicing emergency procedures, such as a pilot's behaviour in case of breakdowns or emergency landings.

Current state-of-the art assumes that simulator training is being carried out by an instructor who leads and assesses the trainee performance, from a customized Instructor Operator Station (IOS) as shown in Fig. 1.

Fig. 1 IOS of the SW-4 simulator used

According to European Aviation Safety Agency (EASA) instructions for teaching in a simulated environment [1], the work of a flight instructor, even in a safe simulated environment, includes a significant amount of multitasking, and monitoring large amounts of data connected to pilot performance. Tasks listed in the EASA instructions are (among others) as follows: prepare resources, present knowledge, facilitate learning, manage time to achieve training objectives, assesses trainee performances, monitor and review progress, evaluate training sessions.

Even focusing solely on the active part of the exercise places a heavy work load on the instructor who must perform professionally and create an appropriate learning environment. Demands on the instructor increase when, in addition to teaching, the instructor must divide his or her attention to log and monitor flight parameters throughout the exercise. Performing these mundane logging and monitoring tasks can divert the instructor's attention and impair the ability to assess pilot performance [2], [3].

To avoid such an effect, the instructor needs suitable tools that reduce the amount of workload, and assist with parts of the exercise not requiring the actual human element to be present. Thanks to the software assist, and implementing some objective criteria of assessment, the instructor's attention could be focused on carrying out more sophisticated parts of the exercise, like communicating with the pilot and leading the exercise. Precisely this idea stood behind designing the Objective Assessment Tool.

The authors created a compact piece of software, integrated with the simulator, providing the instructor with a set of objective flight parameters. The values streamed to IOS assess the precision, reactions, and psychophysiological state of the pilot, during every training exercise. Being provided with such

Maciej Zasuwa, Antoni Kopyt, and Grzegorz Ptasinski are with the Warsaw University of Technology, Faculty of Power and Aeronautical Engineering, Warsaw, Poland (phone: +48 22 234 7515; e-mail: mzasuwa@meil.pw.edu.pl, akopyt@meil.pw.edu.pl, gptasinski@meil.pw.edu.pl).

an information set, in a readable graphic form, significantly reduces the instructor's workload, and his or her attention, can be focused on other parts of the exercise.

Authors recognize the possibility to commercialize the tool as an augmentation of already existing solutions for IOSs at some time in the distant future. However, for the close future, the tool will serve as a learning and development platform for testing algorithm derived for objective assessment of the pilot.

II. Assumptions for Algorithm Development

For the balanced and precisely defined set of requirements used by the embedded algorithm, an ADS-33 [4] report has been used. The ADS-33 report is a document developed by Hoh Aeronautics (21.03.2000), in order to assess handling qualities requirements for rotorcrafts. Since being created, ADS-33 remains one of the most complete works on the subject of rotorcraft handling, not only for U.S. Air Force, but also for civilian aerospace companies not only from USA, but also from rest of the world.

The full report consists of a list of maneuvers proposed for a complex assessment of handling qualities of a rotorcraft, along with appropriate guidelines about how the training courses for the maneuvers should be prepared, and how the maneuvers should be performed.

While the requirements specified in the document are used to describe the handling qualities of the rotorcraft, they need some alterations to pose a good and flexible training ground for developing an algorithm of pilot skills assessment. ADS-33 checks the handling qualities of rotorcrafts, however if criteria are appropriately changed, it can be easily utilized to check pilot's performance during maneuvers in simulated environment.

As all of the imperfections of the simulation model are known to the fullest extent, then all of the errors and faults in manoeuvre trajectory must come from pilot's abilities, lack of attention span, or slow response time. By accounting for the simulator imperfections, and applying a predefined set of manoeuvres, the embedded algorithm can precisely calculate pilot's performance imperfections. The full list of predefined manoeuvres used for the sake of developing the algorithm is presented in the latter part of the paper.

Description of the simulator

The simulator used as the base for developing the embedded algorithm has been fully developed at the Faculty of Aeronautical and Power Engineering of Warsaw University of technology. The device is classified as Flight Navigation Procedures Trainer FNPT. The platform consists of following components [5]: 3-channel visual system, pilot station (helicopter cabin), IOS, PC cluster, responsible for the physical calculations and overall calculation background, software.

The virtual environment of the simulator is projected on the spherical screen via a system of three projectors mounted over the helicopter cabin (see Fig. 2 for reference). The algorithm responsible for the visual environment, connects three flat pictures, in a way which ensures proper projection on the spherical screen. In terms of software, the simulator environment is C++ based, built in accordance with the architecture of HLA. Using only stock PC computers as the simulation engine allows for easy customization and alteration of the model. Suitable MATLAB, Simulink, and LabVIEW open source tools were available for processing and visualizing the flight data obtained by IOS.

Fig. 2 SW-4 simulator used within the project

The visual system consists of spherical screen (allowing field of view of 40 degrees vertically and 180 degrees horizontally), on which an image of the virtual environment is projected. This visualization is created with assistance of a system of three projectors mounted over the helicopter cabin. Images are synchronized and projected with regard to the screen shape (with software assistance), for the sake of eliminating artefacts and deformations.

IOS is located in vicinity, and within visual line of sight from the helicopter cabin. It consists of four 17" monitors, displaying state of the helicopter, map of the terrain, current state of the dials inside cockpits, and several other parameters necessary for appropriate oversight of the exercise performed.

The instructor responsible for the exercise is capable of communicating with the trainee via headsets connected by the TeamSpeak software. By observation of parameters and maps displayed on the IOS, the instructor can fully monitor pilot's performance during exercise. Moreover, using the IOS interface, the instructor is capable of dynamic interaction with the virtual environment. He or she can quickly change weather parameters, as well as simulate variety of emergencies inside the helicopter.

There were no hardware modifications done to the simulator itself. The only alteration done, was mounting the physiological sensors (EKG) inside the helicopter cabin.

During the communication between the pilot cabin and the simulator engine, a frame with data about the state of cockpit is being sent to IOS. From the frame (consisting of 125 various parameters), the authors separated a vector, consisting of nine basic parameters. Values in the vector, allow for spatial identification of helicopter position and orientation, as well as tracking the trajectory performed by the helicopter during exercise. The vector and description of its component variables are shown in Fig. 3.

$$\mathrm{Mref} = \begin{bmatrix} x \\ y \\ z \\ V_x \\ V_y \\ V_z \\ \Theta \\ \Phi \\ \Psi \end{bmatrix} \quad \begin{matrix} \textit{Position of helicopter} \begin{cases} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \end{cases} \\ \textit{Velocities along principal axes} \begin{cases} \boldsymbol{V_x} \\ \boldsymbol{V_y} \\ \boldsymbol{V_z} \end{cases} \\ \textit{Euler angles values} \begin{cases} \boldsymbol{\Theta} \\ \boldsymbol{\Phi} \\ \boldsymbol{\Psi} \end{cases} \end{matrix}$$

Fig. 3 SW-4 state vector

This vector was set as a basis for qualitative assessment of the pilot's performance during each exercise. Values that it contains allow for precise determination of helicopter's spatial state. At the same time, all of the values can be either read from the dials or otherwise tracked by the pilot during the exercise. Therefore, the embedded algorithm assesses the pilot's performance in a "fair" way, taking into account values which are explicitly visible to pilot and are not "hidden".

All of the coordinates used by the algorithm, and listed in this article, are Cartesian and clockwise-positive.

III. Software and Architecture of the System

The architecture of the system was mostly, already included in the simulator itself. The IOS receives a byte array from the simulator physical engine, which consists of basically all flight data, cabin parameters, and information about factors such as weather, time, or amount of fuel left in the helicopter's tanks. All of the tools and software used were kept to bare minimum (open source, or easily accessible tools), for the sake of simplicity and ease of potential modifications.

As the IOS was already able to read the byte array sent from the simulator cabin, the only software modification done was implementing the MATLAB algorithm responsible for the actual mathematical basis of the error calculations, and then showing them in real time on the set of graphs projected onto IOS screen.

In the current state of the project, the authors used only the MATLAB algorithm for both calculating the percentage of error in trajectory and displaying the values in real time on one of the IOS monitors. For better aesthetics, in the coming developments of the project, it is planned to utilize LabVIEW display modules, as they are easily connected with MATLAB scripts.

Fig. 4 is an idea chart that shows basic flow of the data in the system as well as the idea of modularity. The simulation platform provides the embedded algorithm with the data about the attitude and position of the helicopter in the virtual environment. The full set of the data is being registered for after-flight analysis, while the minimal necessary set (the vector shown in the previous section) is used as a base for calculations performed by embedded algorithm. Quality of performance (in terms of total errors of trajectory and altitude) is then streamed to the IOS as a set of easy-to-read graphs, and numerical values. This way, the instructor is being provided with the data necessary for keeping good track of the performance, but at the same time is not overloaded with tasks to execute.

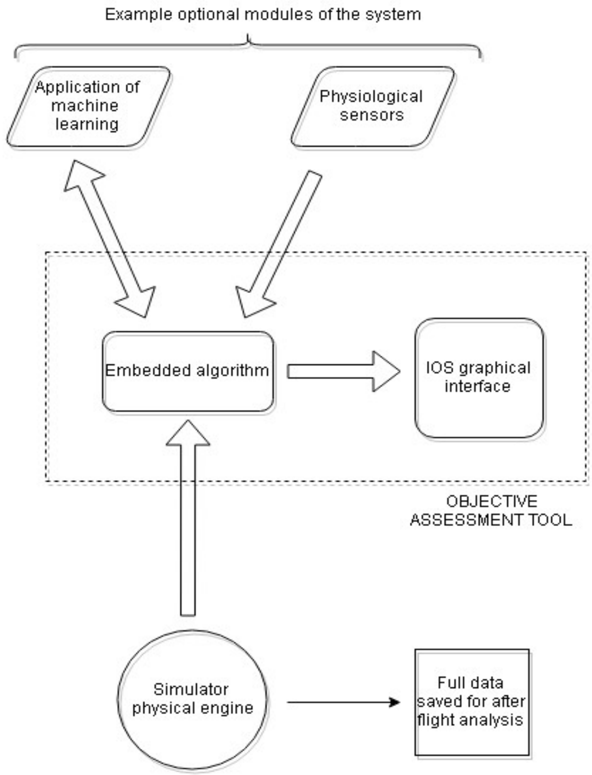

Fig. 4 Flowchart of the designed system

As visible on the flowchart, the idea for the Objective Assessment Tool was to be a modular platform, providing easy implementation and testing of the additional parts and tools. Physiological readings mentioned could prove extremely useful in terms of assessing a pilot's responsiveness and reaction to stress, as well as measuring the trainee's workload. The other planned extension, would be utilization of the data generated by the test flights to try applying self-learning, algorithm, either for data analysis and assessment, or for replicating the flight manoeuvres.

The previously mentioned examples are only two ideas for modules serving as part of the Objective Assessment Tool. As the system is based on the open source solutions, the modularity potential is high and could be developed much further.

As a proof of concept, the authors have proposed implementation of physiological sensor capability. The EKG sensor was proposed mostly for its compact size and ease of implementation, as well as the well-known correlation between human heartbeat and the stress level the specimen is put under [6]-[8].

EKG measurements are widely adapted in aviation, as a tool for quick and easy assessment of stress level and workload of the pilot [9]. By continuous measurement of impulses generated by the human heart, and applying signal analysis, one can assess state of the patient, as well as find any abnormalities [10].

The authors proposed to use a simple USB EKG sensor, mounted on pilot's finger, or on a wrist in a form of bracelet. Both solutions are very compact, and due to the size, do not pose danger of negatively affecting pilot's performance or comfort during the exercise.

The EKG sensor would be integrated into the simulator cabin via an external USB port, and heartbeat information would be included in the data table, already registered by IOS. The embedded algorithm, aside from calculating the errors in trajectory, and monitoring the physical state of the helicopter, would analyze the EKG readings.

Heartbeat would be measured for the rested pilot, and then again after performing a brief exercise to compare calm and stressed levels. Then, concepts from signal theory (e.g., the correlation factor) would be applied to data being gathered during training session to check correlation of the gathered signal with either rested or stressed state Data.

Providing an instructor not only with information about present state of the helicopter, but also about the stress level and potential fatigue of the trainee, could serve a great purpose. This information not only helps with assessment of the performance, but also can improve the quality of training, or even improve such a sensitive and important element as communication between student and the instructor.

IV. Testing and Validation of the System

As part of the design process of the Objective Assessment Tool, a series of tests was carried out. Several test flights of each maneuver were registered to validate the mathematical basis of the algorithm, as well as to check the stability of the application and the Graphical User Interface.

As mentioned before, the simulation platform allows for high degree of customization. This applies also to modelling and implementation of the visual environment projected on the spherical screen. All of the manoeuvres used for the sake of developing the system were modelled in 3DS Max Autodesk software, and then implemented to the virtual environment, after conversion to .ivu format files.

Manoeuvres for implementation in the virtual environment have been chosen based on the ADS-33 report. The full set of manoeuvres have been constructed in a way that tests a variety of skills possessed by trainee, from agility and quick response to sophisticated and delicate manoeuvres.

The six manoeuvres chosen are: slalom, obstacle avoidance in forward flight (following the ground shape), U-Turn, carrier landing, carrier landing with vertical descent, pirouette manoeuvre.

Fig. 5 shows the manner in which manoeuvres were located on the training ground in Virtual Environment All of the manoeuvres consist of visual reference objects (such as pylons and reference trajectories), as well as the contrast surface on the ground (simulating surface of the runway). The whole test course consists of the six test manoeuvres mentioned earlier, as well as the reference trajectories to, and from them.

Such a "comb-like" order of manoeuvres allows the user to perform several training exercises in a smooth, non-disturbed manner. Moreover, by placing manoeuvres this way, it is easy to create simple triggers, launching the appropriate assessment algorithm. When a helicopter flies through a gate localized in certain coordinates, the appropriate algorithm function (based mathematically on the exercise performed) triggers, and starts calculating errors (as well as streaming the graphs onto IOS).

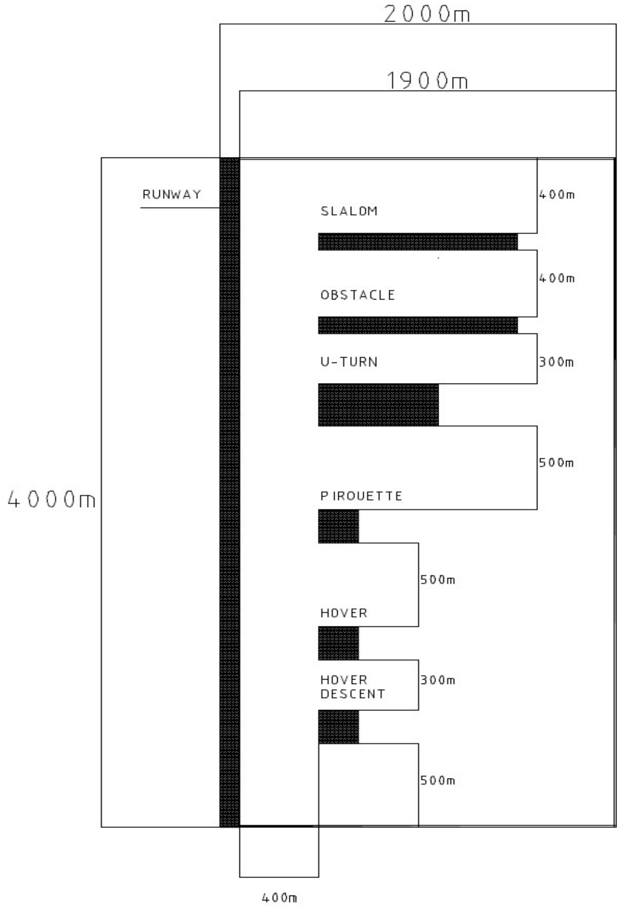

Fig. 5 Maneuvers location in the test course

Figs. 6 and 7 present modelling approach proposed by authors. Fig. 6 shows a direct excerpt from ADS-33 report, while the Fig. 7 visualizes the virtual test course modelled by Authors. The majority of tests in this proof of concept were not performed by professional/military pilots, but by trainees, students, or faculty members. Due to this fact, some auxiliary elements of geometry have been added to make it easier for inexperienced pilots to perform the tasks.

Auxiliary elements for most of the manoeuvres consist of: model trajectory in form of thin 3D pipe, for the pilot to follow; cubical gate showing the desired initial position of helicopter for every manoeuvre; series of rings, leading pilot onto a correct enter trajectory to every manoeuvre.

In addition to the three elements modelled by authors, all of the manoeuvres also consist of elements mentioned by ADS-33, such as pylons and visual reference objects.

Out of six proposed manoeuvres, slalom has been chosen as a representative example of Objective Assessment Tool task realization. Slalom checks a variety of skills, as it needs to be done with certain airspeed, precision, and overall agility for the manoeuvre to be completed safely, but at the same time within boundaries set by the ADS-33 report.

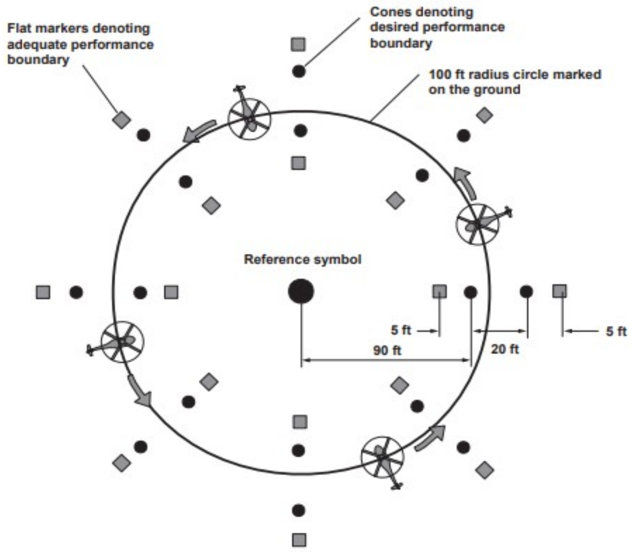

Fig. 6 Pirouette manouevre description from ADS-33 report

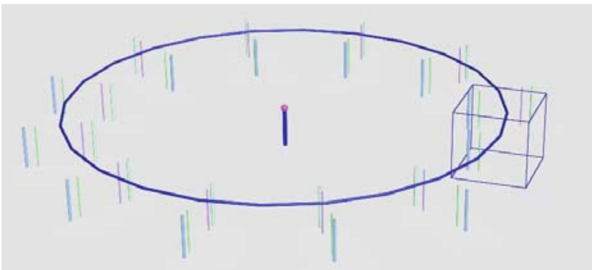

Fig. 7 Pirouette manoeuvre modelled in Virtual Environment, with auxiliary geometry

The quality of the trajectory can easily be seen through visual observation of the graphs generated by the tool. Maneuvers such as the pirouette or hover, however, can be more difficult to assess, especially for inexperienced observers

The exercise was modelled in accordance to ADS-33 report; all of the dimensions and reference objects have been sized precisely as described in the report. In addition to compulsory reference objects described in ADS-33, a model trajectory and series of rings leading to and from manoeuvres has been added, in order to make the exercise easier to approach by an inexperienced pilot. Fig. 8 shows the dimensions of manoeuvre in ADS-33, and Fig. 9 provides shows the 3D model generated in the Virtual Environment.

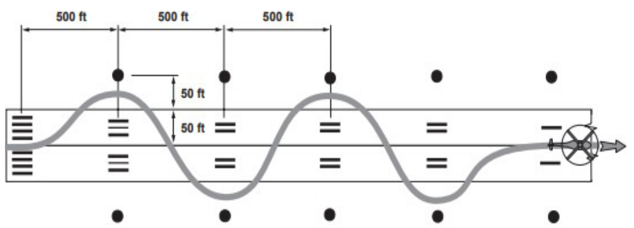

Fig. 8 Suggested course of slalom manoeuvre, according to ADS-33

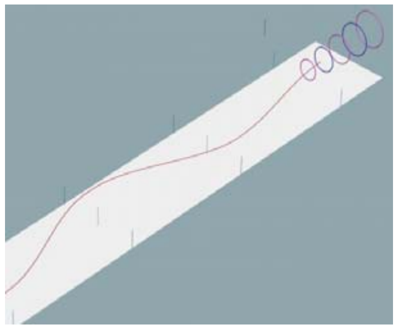

Fig. 9 Slalom manoeuvre modelled in the Virtual Environment

Slalom was modelled in 3DS Max software by adding a 100x1500[m] grey runway (in order to help pilot with identifying visual cues such as pylons), as a background for the test course. Then the set of vertical pylons was added in ranks of 3 (2 on the sides of runway – denoting maximum amplitude of turn – and 1 in the middle), and was placed on a runway every 152.4 meters (500 feet). Pylons denote how the test course should be flown (as in the ADS-33 guide visible on the previous page). Although such a course should be enough according to report specification, some additional visual cues have been added for comfort of the test pilots, namely: set of pylons leading the pilot to the beginning of the course, in order to help navigate to the starting point, and to act as 'gates' for starting the calculations algorithm; similar set of pylons in the end of the manoeuvre course; line in a contrasting color parameterized in such a way, that it shows the pilot "perfect" trajectory for the said manoeuvre. Therefore, not only can the pilot use the pylons to navigate and keep said trajectory, but also can he follow the line.

It is important to note that all of the visual cues (pylons, trajectory line) are modelled in contrast colors (pink, neon blue, neon red) to help the pilot to keep focus on them, but at the same time their size is such, that they are no obstacle for pilot's field of view (FOV), and they cause no distraction.

As ADS-33 report does not provide information about the precise shape of the model slalom trajectory, the authors decided to approach the problem with polynomial approximation, based on the method used by Roberto Celi in [11]. Approximating slalom shape as a polynomial, rather than using trigonometric functions should help retain simplicity, and precision of calculations.

V. Flight Tests Description

The following section presents data gathered during two test flights. The two were picked deliberately in such a way to show the algorithm capabilities to assess pilot performance. Both manoeuvres were performed by the same pilot, the same day. The flights started from the same point on the test course, and were separated by 5 minutes.

The pilot was one of the authors (Antoni Kopyt). He is a– young adult male, without any health issues at the time of testing. Before each test, the pilot was relaxed. While Mr Kopyt had no professional piloting experience prior to the test and his performance errors might differ from the made by trainees, the purpose of the proof of concept was to demonstrate capabilities of the GUI, and calculating algorithm.

To keep the interpretation simple, the authors put only the trajectory graphs in the following section. Other values were omitted in order to make the analysis most understandable, even for inexperienced reader. Fig. 10 provides a general idea of full graphical interface outline.

Depending on the next modules added to the system (e.g., psychophysiological sensors, or algorithm modifications), the graphical interface can be modified for the ergonomic and readable design.

VI. Flight Tests Data Comparison

Data describing trajectories of both flight tests are presented in Figs. 11 and 12 on two consecutive graphs. Below them, numerical values of correlation of trajectory (calculated by embedded algorithm.

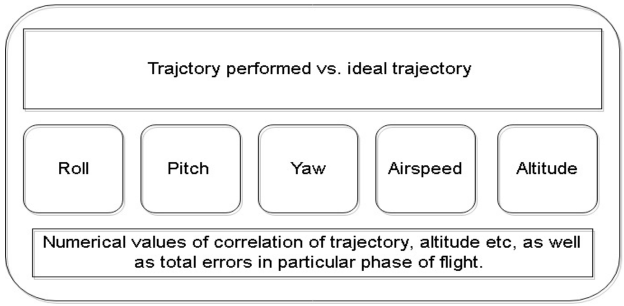

Fig. 10 Proposed GUI outline

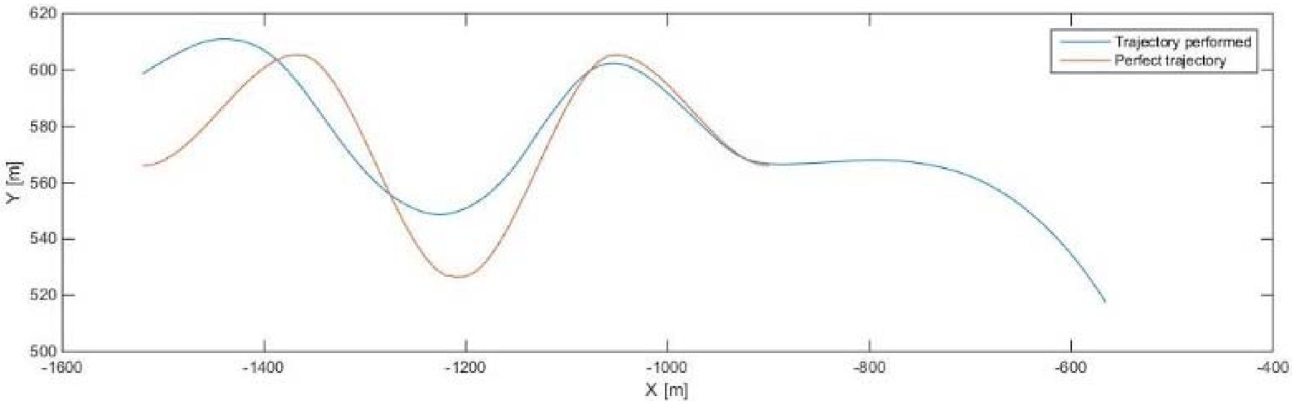

Fig. 11 Trajectory comparison during first flight test

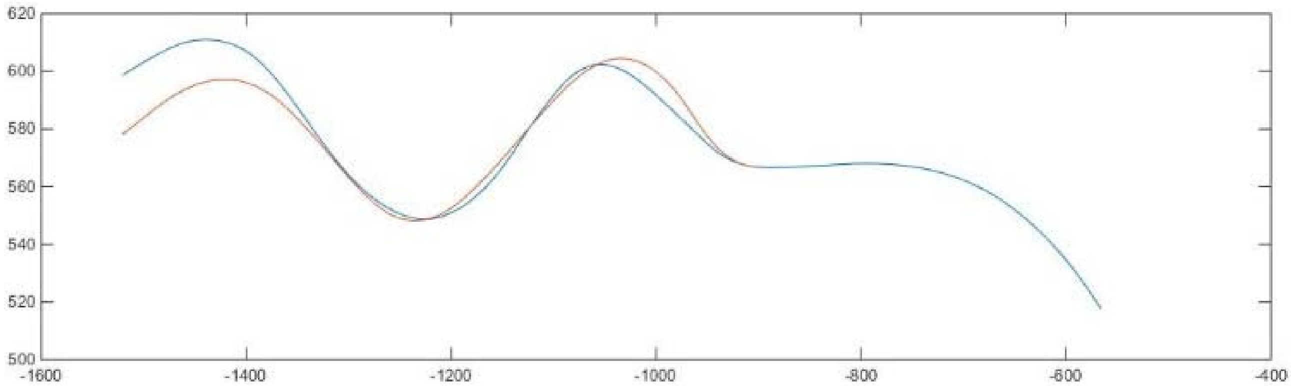

Fig. 12 Trajectory comparison during second flight test

In order to assess precision of pilot's performance during the maneuver, the test course was virtually divided into three parts, in the same manner in which equations (Celi) describe the model trajectory.

For each part of the maneuver the percentage error in trajectory is calculated (as an average value of difference between model trajectories coordinates and appropriate coordinates of the trajectory performed by the test pilot), and presented as a percentage of the slalom amplitude (or other reference value in different exercises). The values in Table 1 describe results of both test flights.

TABLE I
Percentage Errors

	Percentage errors of the manoeuvre performed		
	Part 1	Part 2	Part 3
Test 1	3.84	26.13	40.89
Test 2	8.45	2.69	24.43

It is clearly visible on the graphs showing trajectory, that first manouevre started closer to the model course than the second one. After that, significant error was made in the second part of the exercise. The error continued to the third part.

For the comparison in the second flight test as shown in Fig. 12, the pilot made a slightly bigger error in the beginning (trying to get the appropriate heading for the second turn), then performed the second turn almost flawlessly, only to finish with a quite large error in the end.

The percentage coefficients are presented in Table I. They confirm the conclusions made after visual inspection of Figs. 11 and 12 showing trajectories of both maneuvers. In order to wrap up, average percentage error for both flights was calculated. It is 23.62% for task 1 and 11.86% for task 2.

While calculating percentage error is not a sophisticated method of assessment. In the early phase of project development, it was important to get appropriate data readings, and try to compare them even in a simplest way. For further development authors propose using concepts derived from signals theory.

By using correlation and cross-correlation factors, it would be possible to check similarity of the model trajectory, and the trajectory performed by trainee, as well as other flight parameters (such as Euler angles, or even psychophysiological signals generated by the pilot during performing test exercise).

VII. Summary

The Objective Assessment Tool was designed to serve two important purposes for pilot training. First, it tries to reduce the instructor's workload by providing him or her with a set of easy to read visual data. Second, it calculates and shows percentage coefficients of how correctly each maneuver was performed, compared to the perfect trajectory and physical parameters of flights. Both of these capabilities are planned to be developed during future works on the project.

For the sake of easy implementation, alteration, and possibility to customize the tool for the future user, it was designed using a minimal number of simple and widely available instruments such as Matlab, LabVIEW, or 3DS Max modelling software. Thanks to this, future work on the system should be easy even for new members of the team, not familiar with the tool.

During test maneuvers the authors validated that the algorithm works. This proof of concept case study consisting of two flights performed by the same pilot, verified that the algorithm, in fact, provides data that help to quickly identify flaws in pilot performance without putting much workload on the instructor. Due to this, the instructor is able to focus on more important parts of the training exercise and establish a proper level of communication with trainee, while at the same time easily track his or her progress.

Several opportunities have been identified as paths for future development of the project. Due to high flexibility and level of possible customization, the Objective Assessment Tool can be reprogrammed to help in training of both helicopter and aircraft pilots, as well as UAV operators.

The modularity of the Objective Assessment Tool and the capacity to add peripheral scripts and sensors can expand pilot assessment to include the neural responses. The ease of possible implementation of external sensors (readings of which could be linked to the data table received by IOS), which would help extend the objective assessment capabilities. By measuring psychophysiological parameters of the trainee during performing parameters, it would be possible to also assess the level of stress and workload trainee is put under during the maneuver, and alters the training program accordingly.

The authors recognized the need to apply a more sophisticated assessment model, and to divide each maneuver into individual components, so that the pilot's performance in each part would not affect errors made in the next one. Moreover, for the sake of implementation of psychophysiological sensors, some hardware modifications of the simulator's cabin would be needed (especially exposing the USB ports, and implementing the readings into data frame). Also in order to get the best feedback on the workload reduction, some simulations involving researching psychophysiological parameters of the instructor would be needed.

Process of algorithm development and flight tests resulted in laying important groundwork for the further improvements and alterations of the algorithm. Without proper design, and validation, any further developments would not be possible.

References

[1] EHEST, "Teaching and Testing in Flight Simulation Training Devices (FSTD)," EASA, Koln.
[2] P. J. D. V. Podofillini L., "Measuring the influence of task complexity on human error probability: an empirical evaluation" Nuclear Engineering and Technology, pp. 151-164, 1 April 2013.
[3] L. M. G. A. S. L. M. R. Horrey W., "Distraction and task engagement: How interesting and boring information impact driving performance and subjective and physiological responses" Applied Ergonomics, pp. 342-348, January 2017.
[4] HOH Aerospace, "Aeronautical Design Standard Performance Specification Handling Qualities Requirements for Military Rotorcraft," United States Army Aviation and Missile Command Aviation Engineering Directorate, Redstone Arsenal, Alabama, 2000.
[5] N. J. Zasuwa M., "Integrating Dynamic Models of Mobile Objects with Reconfigurable Simulator," in Proceedings of 5th International Conference Supply on the Wings, Frankfurt, 2010.
[6] K. M. Kohzoh Yoshino, "Correlation between mood and heart rate variability," Health, pp. 553-556, 2017 August 17.
[7] S. Terathongkum, Relationships Among Stress, Blood Pressure, and Heart Rate Variability in Meditators., Virginia Commonwealth University, 2006.
[8] M. Campos, "Heart rate variability: A new way to track well-being.," Harvard Health Publishing, 2017.
[9] Z. K. K. M. A. J. Kwarecki K., "Zmienność Częstości Skurczów Serca u Pilotów Komunikacyjnych w Realnym Locie" Medycyna Lotnicza, pp. 120-121, 1993.
[10] T. U. G. M. F. J. S. D. H. J. Heinze C., "Circadian Rhythms in Heart Rate Measures," in Proceedings of the 6th ESGCO, Berlin, 2010.
[11] R. Celi, "Optimization-based inverse simulation of a helicopter slalom maneuver" Department of Aerospace Engineering University of Maryland, College Park.