SkyVision: A map based Aviation Simulator

Syed Asma Saeed[1], Asif Islam[2], M. Sarosh Umar[3], and Moazzam Siddiqui[4]
[1,2] Students, Department of Computer Engineering, Aligarh Muslim University, Aligarh, India
[3] Professor, Department of Computer Engineering, Aligarh Muslim University, Aligarh, India
[4] Wing Commander, Indian Air Force, India

***Abstract*— This paper introduces SkyVision, a web-based flight simulator made to enhance the flight simulation experience virtually using minimal and light-weight, open-source technologies. Unlike traditional simulators, SkyVision operates entirely through web browser without any extensive data downloads or premium graphics and software. Thus, making it accessible to all the aviation enthusiasts without requiring powerful hardware. By integrating the OpenWeather API for real-time weather information and the MapBox API for 3D maps, our simulator provides realistic and dynamic information about the area. This work provides a cost-effective and simple solution with a user-friendly interface, hence a better alternative to existing simulators by reducing complexity and thus obliterating the requirement for large data sets.**

***Keywords*—3D web map, Flight Simulator, MapBox, OpenWeather, real time weather.**

I. INTRODUCTION

A flight simulator is a specialized training device or software application designed to replicate the experience of flying an aircraft. It offers a simulated environment that includes the representations of control, instruments, and operational behavior, of the real aircraft. From pilot training to developmental test of aircraft, even entertainment purposes flight simulators have been very much useful in many applications. They offer a completely safe and cost-effective method through which pilots can train in their ability to fly, exposed to all kinds of scenarios and conditions without the necessity of real flight. Advanced features such as realistic flight dynamics, accurate simulations of weather, and detailed terrain exploration, have become crucial to create these experiences as immersive as possible for the user.

The fundamental concept behind flight simulators is to recreate the experience of flying an aircraft in a highly realistic and immersive manner. This involves not only the dynamics of flight but also real responses of an aircraft under various conditions: temperature, turbulence, windshear, air pressure, clouds, precipitation, fog, and the like. Primarily, flight simulators are used for training, aircraft design and development, as well as the research of aircraft characteristics and control handling qualities.

The existing simulators are made based on generally two approaches. The first approach uses data streaming where the data (imagery, elevation, terrain, etc.) is loaded dynamically to the users' PC as and when required. The data of the location chosen by the user are transferred to their PCs through streaming for the simulation process. Suppose, user A wants to fly over Chicago, the data of Chicago is first transferred to the user's system before they can begin the simulation. Due to the cache limitation of the browser, the data of the distant objects are not transferred when streaming aims to balance data transfer efficacy and visual quality. These kinds of simulators may either use raster tiles or vector tiles for streaming data. become pixelated when zoomed in, as the resolution is Raster tiles are, in essence, fixed images, that is, data is represented in terms of grid of pixels, and each pixel has a specific color or value, making them ideal for detailed imagery like aerial photographs. Yet, raster tiles are limited to the fixed pixel grid. Whereas vector tiles represent data with geometric shapes, that is, points, lines, and polygons. Unlike raster tiles, they are resolution-independent, meaning that they can scale to a really good level without the loss of clarity. They are, therefore, generally more flexible for rendering maps and terrain as the details can be very dynamic in respect to zoom levels. However, vector tiles require more processing power to render the shapes and are often used for more abstract or stylized map representations.

The second approach uses predefined terrains; these simulators are built for specific terrains and are unable to explore terrains other than the predefined ones. Suppose, a simulator is designed for Vienna, then a user who wants to fly over Chicago cannot do so. These kinds of simulators are associated with the limitation of the geographical area they cover.

In this work, the proposed system is designed to keep the simulator as simple as possible by using basic technologies to eliminate the need for downloading anything on the user's PC, thereby saving both space and data on the user's side. The core idea involved is to use open-source software as the underlying architecture to replicate the real-world flight. For this, we have used MapBox GL JS API to generate the terrain and the OpenWeather API to provide all the relevant weather information about the flight. The implementation relies on JavaScript to ensure browser-friendliness. This architecture is chosen to keep the system cost-effective and robust. To use the proposed application, the user only needs a browser and good internet connectivity.

II. LITERATURE REVIEW

Literature on flight simulation emphasizes its role in pilot training. Effective training modules covering take-off, landing, navigation, and emergency procedures contribute to skill development. One of the critical challenges faced is the loss of control (LOC), a major factor contributing to 41% of aircraft crashes, often stemming from challenges in the clarity of aircraft controls [5]. This alarming statistic highlights the pressing need for enhanced pilot training, particularly in addressing rotorcraft loss of control in-flight (LOC-I) accidents. Research in this area emphasizes the importance of high-fidelity flight simulators, which are instrumental in improving pilot preparedness and reducing accident rates.

The evolution of flight simulation technology, from the rudimentary flight trainers of the early 20th century to the sophisticated full-flight simulators of today, showcases its vital role in both commercial and military pilot training [6]. Additionally, simulators have become indispensable in aircraft design and engineering. Despite persistent challenges with data availability, component sourcing, and the lack of early industry standards, the simulation industry has gained substantial credibility. This growth is largely due to the concerted efforts of airlines to establish common standards, which have led to widespread regulatory acceptance of simulator-based training. The industry's expansion into a multi-million-dollar sector can also be attributed to pioneers like Edwin Link, often referred to as the "Father of Simulation" [7].

Recent studies have explored the integration of web maps into flight simulators. For example, a basic flight simulator using Silverlight 3 and Bing Maps 3D data was developed to provide a 3D environment and aircraft cockpit experience. This approach enables the use of low-cost PCs as flight simulator clients, accessing extensive simulation data stored on Bing Maps servers in real time[4].

Different flight simulation tools have different strengths and limitations for flight control co-simulation with Matlab/Simulink [6]. This research provides a comprehensive review and tutorial on using Matlab/Simulink for flight control co-simulation with various popular flight simulators to evaluate aircraft models and control systems, discuss the advantages and limitations of each approach, and demonstrate the use of software-in-the-loop (SIL) and hardware-in-the-loop (HIL) simulation for small UAV systems.

Another study presents a method for converting 2D geological maps into 3D visualizations using the Google Earth virtual globe [8]. By applying GIS techniques, the research analyzes the geodiversity of natural areas such as Spain's Arribes del Duero Natural Park and Quilamas Natural Area. Additionally, an augmented reality (AR) prototype designed to improve spatial perception in flight training simulators is discussed, highlighting key design elements like ground projections, grid overlays, and intuitive GUI placement [9].

When taking into account some specific flight simulator, one will find Microsoft Flight Simulator [10], which is known for its realism and detailed graphics. That gives flying a live atmosphere because of applications of real weather conditions, very detailed scenery, and therefore, making it a favourite among both casual users and serious aviation enthusiasts.

X-Plane [11] is said to have good and detailed flight dynamics, having a wide range of aircrafts. It is an open platform that lets every user tweak and enhance his experience, which makes it go-to for those who desire a high level of customization and realism while flying.

Aerofly [12] offers a user-friendly experience with smooth performance that is perfect for beginners and experienced pilots both. It brings ease of use with a range of aircrafts as well as locations, which make it more than just another great choice for any potential user who wishes for a kind of approachable yet immersive kind of flight simulation experience.

Infinite Flight [13] delivers flight simulation onto your fingertips, giving you a good simulation directly in your mobile device. It gives users global scenery and range of aircraft in order to enjoy the reality of flying straight from their phone or tablet.

The most immersive, Google Earth [13] allows the user to explore satellite imagery and 3D terrain all around the world. Its deep zoom and easy navigation features make it visually easy to view detailed landscapes, cities, and landmarks. Not being specifically a flight simulator, its detailed and interactive maps can make it a useful tool in virtual exploration and planning. This provides a unique way in visually understanding a different location and terrains.

III. BACKGROUND AND RELATED TERMS

Understanding the fundamental motions of an aircraft (pitch, roll, and yaw) is thus crucial in addressing these issues. These movements of the aircraft operate on the basis of the movements around the prime axes of the aircraft viz Longitudinal, Lateral and Vertical as illustrated in Figure 1. All these axes meet at a point called the Centre of Gravity.

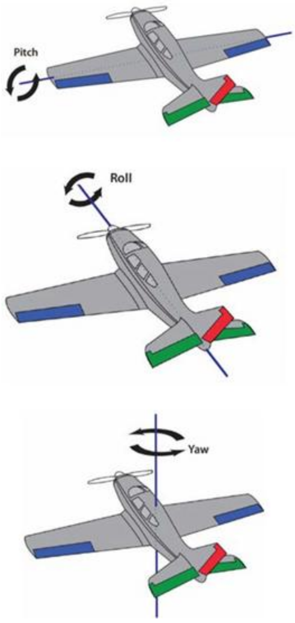

Figure 1: Flight Movements [14]

Pitch- Pitch refers to the upward or downward movement of the aircraft's nose in order to make the plane ascend or descend.

Yaw- Yaw refers to the aircraft's nose movement to point away from the direction of the turn, thus enabling it to turn in a left or right direction.

Roll- Roll refers to the aircraft's movement where one wing rises while the other lowers. This occurs along the longitudinal axis, allowing the aircraft to tilt from side to side.

IV. MATERIALS AND METHODS

A. MapBox

MapBox [1] GL JS is a client-side JavaScript library for building web maps and web applications. With MapBox's modern mapping technology, one can use it to display MapBox maps in a web browser or client, add user interactivity, and customize the map experience in the application. It can be used for visualizing and animating geographic data, querying and filtering features on a map, dynamically displaying and styling custom client-side data on a map, 3D data visualizations and animations, etc.

B. OpenWeather

OpenWeather [2] provides various information related to weather by providing different APIs with different levels of information. The information of the API includes current weather conditions, forecast, maps and different services.

C. Dataset

Our proposed system leverages a dataset [3] of Indian airports to provide users with details of the nearest airport during flight. The original dataset consists of 345 entries and 20 features. After preprocessing, the data was refined to 317 entries and 6 key features, with unnecessary information removed to ensure consistency and relevance.

V. PROPOSED METHOD

In this paper, SkyVision: A Map-based Flight Simulator, a flight simulation is designed to get beyond the limitations of the existing simulators such as restricted geographical domain, no real time weather updates and static view of the map. Our system overcomes these drawbacks and constraints while also considering the high subscription cost of the existing simulators and choosing open-source architecture to reduce the cost.

The registration process is the first step in this system's user-friendly experience. After providing valid login credentials, users can log in to the system and become members. Then they are granted access to a personalized dashboard that displays their information, accomplishments, and awards.

SkyVision leverages the MapBox API to retrieve terrain data and the OpenWeather API to gather real-time weather information. These API responses are integral to the functioning of the application, providing the foundational data for the simulation. The entire workflow is illustrated in Figure 2.

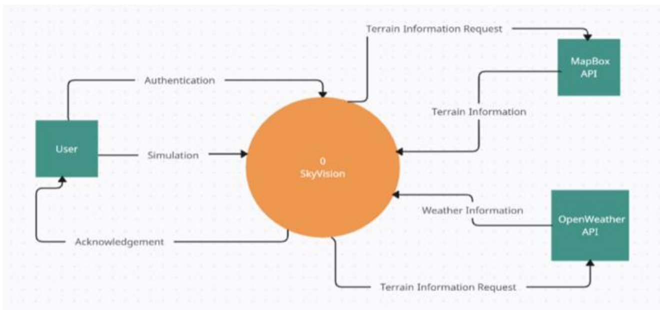

Figure 2: Data Flow Diagram

The API of MapBox GL JS provides the map with the initial configurations with the normal zoom level of 16, a pitch (head of the aircraft), bearing (angle of the aircraft) of 0 degrees with the satellite-streets view toggled. The MapBox Geocoding API centers the map to the entered location by fetching the coordinates. On the backend it uses the flyto() functon to position the map at the located coordinates with the same zoom level i.e 16.

Pressing the Arrow Up key causes the getCenter() function to continuously retrieve the map's current coordinates, simulating the aircraft's forward movement. We then modify the longitude by subtracting a delta of 0.01 to obtain the updated position and weather data. With the delta distance set to -100, the panBy() function is used to pan the map. Figure 3 lists the different keys utilized for the movements. For pitching the map downward, we retrieve the current pitch using the getPitch() function, and increase it by 1 using the setPitch() function. Similarly, to pitch the map upward, we decrease the pitch by 1.

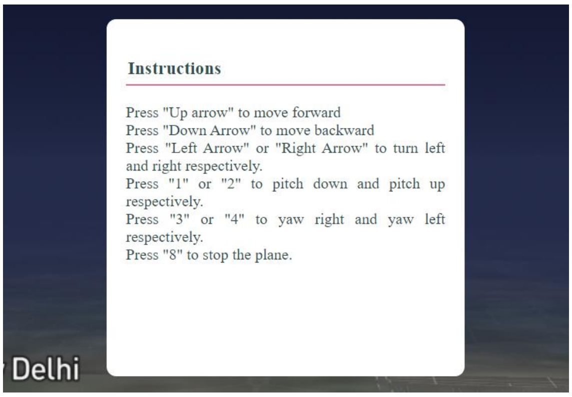

Figure 3: Instruction-key menu

The inbuilt getBearing() function is used to retrieve the current bearing and then we can yaw the map left or right by by adding a predefined value of 25 degrees.To roll the map, pitch and bearing, both needs to bae adjusted simultaneously as described in the figure above.

For the information of the location, we use the reverse geocoding feature of the MapBox API where the coordinates are utilized to find the exact city and the country name. This information is then also used for the text-to-speech synthesis and the JavaScript audio feature providing the vocal information of the place while flying.

Weather information as mentioned above is retrieved from the API of OpenWeather displaying different information such as wind speed, temperature, humidity.

VI. Results and Discussion

The proposed system is a complete web application to experience the flying experience using lightweight tools such as MapBox and OpenWeather as the underlying architecture behind SkyVision.

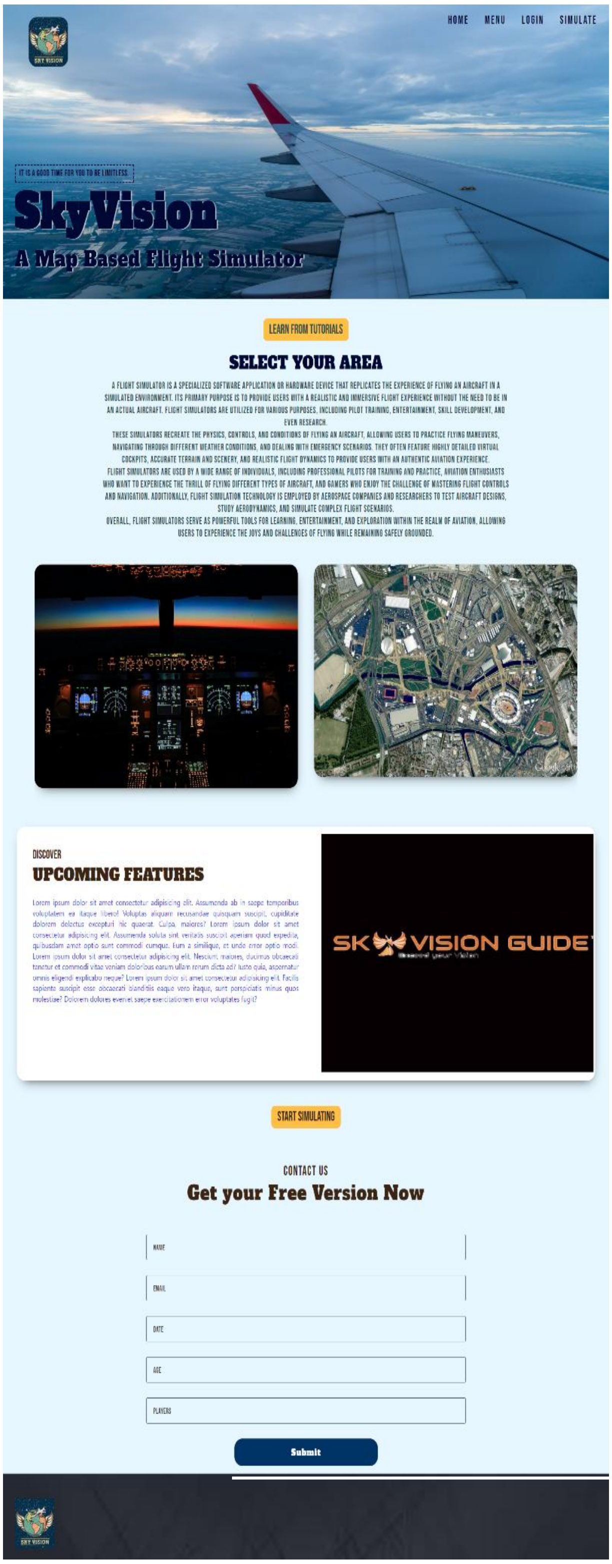

Figure 4: Home Page

The Figure 4 above is of the home page of the proposed application from where the users can access different features of the system.

The login credentials of the users are stored in the MongoDB database and fetched while logging in as shown in Figure 5 and Figure 6.

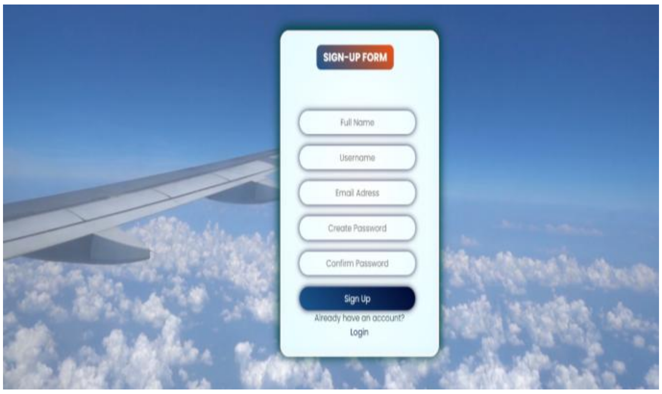

Figure 5: Sign-up Page

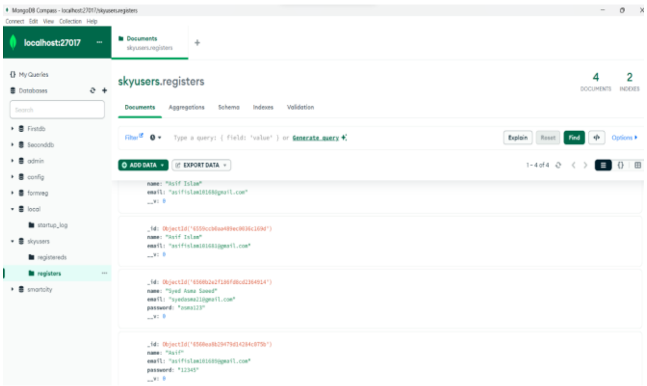

Figure 6: Mongo-DB Database

This system feature allows users the flexibility to select any location of their preference for their simulated flight. Instead of being restricted to predetermined scenarios or fixed routes, users have the freedom to choose the geographical location they wish to fly over within the simulator. This feature enhances the user experience by providing a personalised and customizable aspect to the simulation, allowing individuals to explore and navigate virtual airspace over locations that hold personal significance or appeal to their aviation interests. Whether it's flying over iconic landmarks, exploring specific terrains, or practising manoeuvres in diverse environments, this provision adds a dynamic and interactive dimension to the simulated flying experience. flyTo() function in the mapbox does this operation in the system.

A distinctive feature of our system that makes it more reliable and realistic is that it provides the necessary information required for flying an aircraft in real life such as weather conditions like humidity, wind direction, speed, altitude, bearing of the aircraft, latitude, longitude of the nearest airport and many more. SkyVision uses Mapbox, the OpenWeather API to make system look super realistic and give real experience for any flight simulation to the user as depicted in Figure 7. The addition of real-time weather from the OpenWeather API is used to provide necessary information that a pilot requires while flying about the weather. This powerful combination of technology ensures that the system provides a realistic, immersive aviation experience in an environment where terrain details and weather conditions are depicted accurately.This information somewhat makes the pilot more informed and confident while training in real-life scenarios.

Figure 7: Simulator-Page

A personalised Dashboard in any application serves an important purpose of user engagement and motivation. Moreover, it also gives satisfaction and some identity to the user while using the application whether it is a game or a professional application. Thus, the dashboard acts as a central hub to track the progress, see any message, connect with people, personalise their content and become more immersive and connected to the system as shown in Figure8.

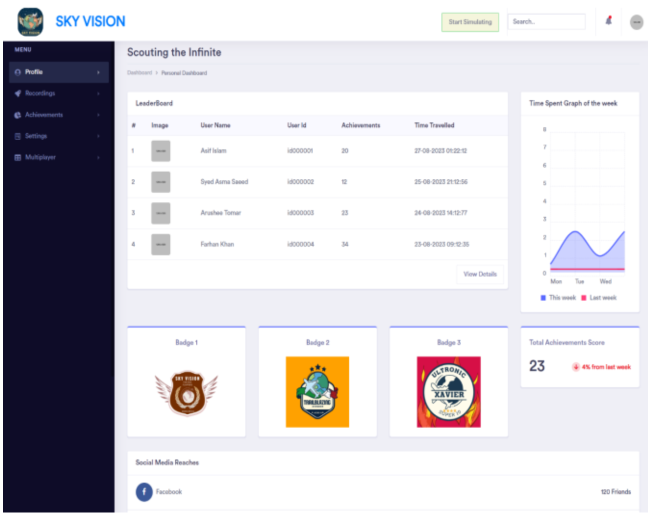

Figure 8: User Dashboard

There are different sections on the dashboard where the user gets the chance to record their sessions. SkyVision provides its users with the feasibility to record their journey so that they can use it for future references and betterment. This feature allows users to capture the entirety. The users can also see their achievements, certificates and time spent while flying the simulator. Though many of the features in this dashboard are not in function, it has a huge scope to be scalable and be more interactive.

The system also has a good feature to get feedback and suggestions directly related to the user engagement. Developers can interact with users who have their ideas and recommendations on the discussion, bringing out a user-centric environment. This allows the user community to contribute additional improvements to SkyVision, turning it into a simulator that meets theirs and our needs.

The motive for the development of this application since the beginning has been user-friendliness and efficient usage of resources which is why there is no downloadable data in it. Every bit of information is being loaded through the internet and the only constraint in the system is the availability of a good internet connection.

CONCLUSION

By using MapBox and OpenWeather as the underlying architecture behind SkyVision, the proposed work ensures to deliver an aviation simulator free from extensively costing elements while delivering the services required in an ideal simulator. SkyVision, being the result of a profound analysis of the limitations in the current aviation systems, promises to deliver a simulator which provides an authentic and realistic virtual flying experience. During the formulation of SkyVision, the main focus and emphasis was laid on simplicity. From the beginning till the end, it was kept in mind that the user should not get overburdened and find the steps self-explanatory and simple to go ahead with the simulator.

At the heart of SkyVision is a commitment to address the deficiencies inherent in current aviation systems. We prioritize realism, accessibility, customization, and seamless integration with real-world data, including satellite imagery. Tailored for aviation enthusiasts, pilots, and hobbyist aviators, SkyVision offers a virtual piloting experience with a focus on fighter aircraft. Dynamic weather elements coupled with day and night modes create an environment that mirrors actual flying conditions. Real-time Flight information that is required for smooth and uninterrupted aviation is displayed on the screen like altitude and bearings which serves a major role in operating an aircraft. Similarly, the direction of the wind which is an important factor to decide several considerations for the pilot is also being provided. Weather information like wind speed and humidity along with different weather modes makes our simulator versatile in terms of diverse and realistic experience. Additionally, knowing the coordinates of the closest airports helps the pilot when they are in the air because it is a necessary feature for any aircraft to know which station to land at in an emergency.

By tackling the major problems encountered in aviation industry simulators, that is, the constraint of limited terrain, cost and frequent consumption of data with greater bandwidth requirement, SkyVision makes sure to deliver a lightweight system satisfying the basic requirements of a simulator.

REFERENCES

[1] Mapbox, "Mapbox GL JS API documentation," *Mapbox*. Available at: https://docs.mapbox.com/mapbox-gl-js/api/ (Accessed 19 August 2023).

[2] OpenWeather, "OpenWeather API documentation," *OpenWeather*. Available at: https://openweathermap.org/api (Accessed 19 August 2023).

[3] OurAirports, "Our Airports Dataset," *OurAirports*. Available at: https://data.world/ourairports/5645d365-f8ec-434b-99ad-fc738d40b0de (Accessed on 19 August 2023).

[4] A. Maiti, S.I. Ao, C. Douglas, W.S. Grundfest, and J. Burgstone, "Flight simulation in 3D Web Map Services using Silverlight™ 3 to Reduce Simulation Costs in Pilot Training Institutes," *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, 2009.

[5] M.D. White, G.D. Padfield, L. Lu, S. Advani, and M. Potter, "Review of flight simulation fidelity requirements to help reduce 'rotorcraft loss of control in-flight' accident rates," *CEAS Aeronautical Journal*, vol. 12, no. 4, pp. 701-721, 2021.

[6] N. Horri and M. Pietraszko, "A Tutorial and Review on Flight Control Co-Simulation Using Matlab/Simulink and Flight Simulators," *Automation*, vol. 3, no. 3, pp. 486-510, 2022. https://doi.org/10.3390/automation3030025

[7] E. Helijah, *Principles of Flight Simulation*. Available at: http://helijah.free.fr/dev/Principles-of-Flight-Simulation.pdf (Accessed: 18 August 2024).

[8] A.M. Martinez-Graña, J.L. Goy, and C. Cimarra, "2D to 3D geologic mapping transformation using virtual globes and flight simulators and their applications in the analysis of geodiversity in natural areas," *Environmental Earth Sciences*, vol. 73, pp. 8023-8034, 2015. https://doi.org/10.1007/s12665-014-3959-1

[9] B. Bach and T. Alexander, "Design and Evaluation of a 3D Map View using Augmented Reality in Flight Training Simulators," *Diva Portal*, 2018. Available at:

https://www.diva-portal.org/smash/get/diva2:1252116/FULLTEXT01.pdf

[10] Microsoft Flight Simulator. Available at: https://www.flightsimulator.com/. Accessed on 6 September 2023.

[11] X-Plane. Available at: https://www.x-plane.com/. Accessed on 6 September 2023.

[12] Aerofly. Available at: https://www.aerofly.com/. Accessed on 6 September 2023.

[13] Infinite Flight. Available at: https://infiniteflight.com/. Accessed on 6 September 2023.

[14] NASA, "Axes and Control Surfaces," *NASA*. Available at: https://www.nasa.gov/wp-content/uploads/2023/06/axes-control-surfaces-k-4.pdf. Accessed on 5 September 2023.