Mixed Reality-Based Simulator for Drone Pilot Training

Suhas G K[1], Dr Yogita[2], Vikas B O[3], Ms. Ritu Dagar[3], Mr.Bala Namasivayam[4],
Prof. (Dr.) Shashi Kant Gupta[5], Dr Bhagappa[6], Prof. (Dr.) Prakash Kuppuswamy[6]

[1]PDF Researcher, Lincoln University College, Malaysia, Sir M Visvesvaraya Institute of Technology, Bengaluru.
[2]Department of Information Technology, BPIT, Rohini, Delhi, India.
[3]Research Scholar, Sir M Visvesvaraya Institute of Technology, Visvesvaraya Technological University, Bengaluru.
[3]Department of Computer Science and Engineering, SRM University, Sonepat, Delhi-NCR, India.
[4, 6]Department of Computer Science Engineering, Koneru Lakshmaiah Education Foundation, Vadeeswaram, A.P. India.
[5]Lincoln University College, Malaysia, Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology. Chitkara University, Rajpura, 140401, Punjab, India.
Department of CSE-AIML, Sapthagiri NPS University (SNPSU), Bangalore, india

Corresponding Author
Dr Suhas G K
PDF Researcher, Lincoln University College, Malaysia,
Sir M Visvesvaraya Institute of Technology Bengaluru
Pdf.suhas@lincoln.edu.my

Abstract
Background: Due to their ability to do a variety of tasks more quickly and effectively than traditional methods, drones have become more and more popular in recent years. Drones are used in many different industries, such as agriculture, surveying, photography, and search and rescue. However, piloting a drone requires specific knowledge and training due to the advanced nature of the technology. **Materials and Methods:** Drone pilot training can be challenging and costly because it calls for practical experience as well as knowledge of safety regulations. One possible solution to the challenges of drone pilot training is the use of simulators. Operators can refine their skills and gain experience in a safe, controlled environment without worrying about damaging their equipment or injuring others thanks to simulators. Creating an immersive drone flight training system that blends in smoothly with the actual environment and gives users a higher level of involvement without motion sickness is the current problem. **Results and Discussion:** For authenticity, the simulation must use physics that accurately replicate the dynamics of drone flying rather than using fictitious models. Implementing a user interface that provides precise telemetry readings is essential to achieving this goal since it gives pilots a thorough grasp of important realistic factors while in flight. In order to ensure expertise in drone operation through real-world scenarios, the system also requires an advanced training module that conveys practical skills and information. Drone pilot training is hampered by the limitations of simulators, such as their resemblance to virtual reality games rather than genuine training settings. Using programs like Unity3D and Blender, we meticulously model the virtual world to replicate the real environment in order to address these issues. Furthermore, using RPM data from the open-source flight planner ArduPilot, attempts

have been made to replicate the drone's physics by adding static and dynamic propeller thrust. By allowing users to alternate between different viewpoints, such as first person, third person, and fixed views, this effort also seeks to provide users' control. A hoop course system was used in the testing process. This required a proficient drone pilot to navigate through more demanding courses, which were made more difficult by elements like smaller hoops, more complex maneuvers, more hoops, and less lighting. **Conclusion:** The benchmark was the pilot's score and timing. The program was then tested by inexperienced pilots, and their times and scores were noted. The course became more challenging as they got closer to the standard.

Keywords: ***Drone Pilot, Augmented Reality, Virtual Reality, Mixed Reality, ArduPilot, Simulation etc.,***

Introduction

This paper presents a Mixed Reality drone simulator, driven by the need to offer secure, affordable, and entertaining environments for operating actual drones. Understanding this endeavor's larger context and related terms is the goal of this undertaking. Introduction to Mixed Reality Users' interactions with computer-generated environments have been profoundly altered by virtual reality (VR) and augmented reality (AR). AR incorporates digital content into the physical world, whereas VR immerses viewers in virtual environments. However, AR is hampered by the obvious division between the real and virtual worlds, which also lessens user immersion. [5]. By smoothly fusing the actual and virtual worlds, Mixed Reality (MR) solves this problem. It depicts a spectrum where computer-generated content is mixed with an individual's experience of the real world [4]. This is achieved using a dynamic interface that enables interaction between virtual and physical items. [6]. This enables the user to engage with his environment normally while also providing a dynamic and engaging user experience [5]. The benefit of VR and MR is the higher degree of immersion wherein users are free to move their heads around without any limitations on how they perceive their surroundings. [1]. Additionally, the resolution is thought to be superior to that of computer screens, VR headsets made for the same purpose, and printed rendered images. [3].

Objectives

1. **Virtual World Modelling**
   - Develop a virtual environment replicating real-world scenarios.
   - Implement realistic features and landscapes within the virtual space.

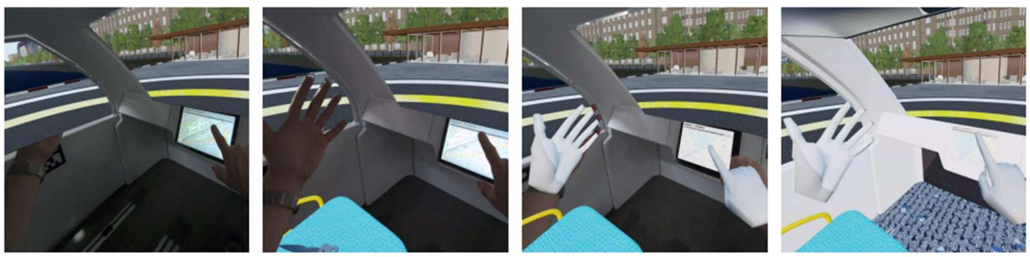
2. **Drone Physics Simulation**
   - Construct a comprehensive physics model for an accurate drone simulator.
   - Ensure the simulation mirrors the behaviour of real-world drones.

3. **Integration with ArduPilot**
   - Establish seamless connectivity with ArduPilot for real-time interaction.
   - Enable the virtual drone to mimic the actions and responses of an actual drone.

4. **Mixed Reality Implementation**
   - Integrate mixed reality features to provide a visual transition between the real and virtual worlds.

Literature Review

Mixed Reality Training Systems in Aviation enables projector users, remote experts, and drone players to engage in shared virtual flying scenarios. Interestingly, bidirectional communication is made possible by the integration of Ultra-Wideband (UWB) sensors, which gives drone players control over virtual components. Potential uses in a range of sectors, such as entertainment, defence, and firefighting, are envisaged by the study. Subsequent investigations endeavour to appraise the system's usability via user evaluations, tackle detected challenges, and enhance the system according to professional advice [10]. Moving on to air taxi simulation [1] an extensive investigation conducted involved the construction of four different mixed reality environments. The level of immersion varied among these setups, and Varjo's marker tracking technology was essential in keeping the virtual cabin model and the real cabin mock-up in line. The evaluation considered variables like the total rating, user interface interaction, and a comparison of MR setups between virtual and actual cabin scenarios. Notably, compared to the real and mixed setups, participants thought the virtual cabin setup was more immersive. The study offers important insights into the effectiveness and user experience of various MR configurations for air taxi simulation. It made use of the Varjo XR 3 and a custom air taxi mock-up. It sheds light on the benefits and drawbacks of each configuration, especially regarding user interaction, immersion, and the realism of the flight simulation [1].



(a) RC (b) MC_{RI} (c) MC_{VI} (d) VC

Figure 1. The four different mixed reality setups presented on the HMD. In the RC setup (4a) most of the environment is real and only the out-the-window view is virtual. At the other end of the spectrum, the VC setup (4d) shows a fully virtual environment. MCRI (4b) and MCVI (4c) represent intermediate between these two extrema.[1]

The XR-OOM system was presented by [11] in the field of mixed reality driving simulation. It smoothly combines virtual and physical components to provide realistic driving simulations. A real car, a Varjo XR headset, and metrics on driving behaviour, presence, engagement, and overall satisfaction are all included in this immersive system, which analyses user performance. High levels of user engagement and satisfaction are demonstrated by the XR-OOM system, which also responds quickly to user actions. A performance evaluation of the system reveals robust sensor and XR components, excellent usability, minimal latency, high accuracy, and seamless integration with current tools. Evaluations based on particular situations show how different design approaches work to accomplish goals like increasing safety and decreasing driver distraction [11,16]. With an emphasis on drone flight control, [12] suggested an interactive training system that pairs a remote trainer with a drone player in a dual-role interaction. Without the drone player present, the trainer which has a head-mounted display (HMD) conducts remote flight control training scenarios. Drone players can operate first-person-view (FPV) drone flight control in open areas with few obstacles by creating and adjusting training scenarios with ease using the Scenario Timeline & Attention Dock (STnAD) user interface. The drone player and trainer can communicate easily through voice thanks to Photon Unity Networking and Photon Voice. To further enhance mixed reality rendering during the drone player's experience, the system incorporates the ZED software development kit to provide relative drone coordinates, orientation, and 3D depth [12, 17]. Examining the possible advantages and disadvantages of extended reality (xR) technology in the context of flight training and aviation [13]. The study uses a Likert scale to measure participants' comfort level with virtual reality (VR) and the environment's applicability to flight training. It collects performance, cognitive load, and eye-tracking data in both virtual and real-world settings. Procedure task completion and altitude and airspeed control performance show notable improvements following 1.5 hours of Virtual Reality Learning Environment (VRLE) training. According to the findings, immersive virtual reality training could be a useful alternative to drone operator training for reskilling and forecasting human performance in real-world flying scenarios [13]. Finally, [8] adds to the conversation by evaluating flight deck training exercises for pilot procedures and skilled drone teleoperation using virtual reality goggles and 2D monitors. Time completion and accuracy are used in the study as performance metrics, and they are graded on a three-point system. In addition, usability and comfort metrics are considered, with mean values displayed next to the student's p-values. The findings show that when using the Head-Mounted Display (HMD) instead of the Desktop Monitor (DM), users were able to locate instruments on the Cessna dashboard more quickly, saving an average of 64% of the time. In addition, the HMD performed with more accuracy than the DM. The checklist took about the same amount of time to finish, but the DM was far more accurate [8].

Figure 2. The cockpit dashboard (Cessna 152) [8]

Proposed System

This proposed model provides a thorough explanation of the architecture of the system, including the frameworks utilised, the physical setup, and the integration of the ArduPilot framework with the Unity game engine for realistic drone physics. This part also explains how the physics of a drone work, as well as the various manoeuvres and how they are recreated in our system. Finally, it discusses how this arrangement would be used for training, as well as numerous scenarios.

Architecture

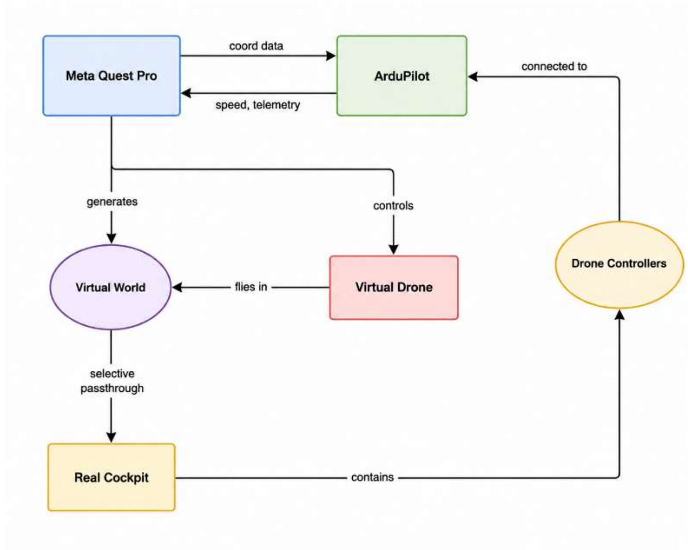


Figure 3.1: Architecture of the proposed system

The user is situated inside a fake cockpit that has a drone joystick controller, as shown in Figure 3.1. Through a USB connection, this controller is linked to the ArduPilot Software in the Loop (SITL) that is operating within the Mission Planner application. The RC channels of Mission Planner, which continuously gathers data from the joystick, are mapped to the joystick controls.The Meta Quest Pro headset's Unity program connects wirelessly via UDP to the ArduPilot SITL. This application allows users to fly a virtual drone in a virtual world. Additionally, the headset displays a real-world passthrough of the cockpit. The ArduPilot SITL sends a JSON object to the Unity application. The computed Pulse Width Modulation (PWM) values for each motor are contained in this item. The virtual drone can hover or move in a specific direction when the application applies these values to its four motors. The ArduPilot SITL receives another JSON object from the Unity application in exchange. The position, speed, acceleration, and other characteristics of the virtual drone in flight are all contained in this item. Continuous data interchange between the ArduPilot SITL and the Unity application enables real-time control and feedback of the virtual drone's flight.

3.1 Configuration of simulation environment

ArduPilot

ArduPilot is an open-source autopilot firmware that works with a variety of vehicles, including multi-copters, rovers, and helicopters. It is commonly used on hobby UAVs and may be combined with ground control software to enable additional functionality such as real-time communication. The firmware employs a controller that receives input from the vehicle's sensors and sends output to devices like as the ESC (Electronic Speed Controller), servos, and so on.

In a physical setup in which ArduPilot is operated on a flight controller, the input to the flight controller is provided by a radio receiver with at least four channels for Throttle, Yaw, Roll, and Pitch inputs, as well as optional auxiliary channels. The sensory data is then used by ArduPilot to do complicated mathematical calculations, such as the usage of Karman filters and PIDs, to calculate the proper individual motor speeds to stabilize and propel the vehicle. It then communicates with the ESCs (Electronic Speed Controllers) via PWM (Pulse Width Modulation), a technique used to manage the average power given to a load, in this case a brushless motor, by varying the width of the pulses. This entire procedure is done indefinitely in a loop.

ArduPilot can be integrated with a Ground Control Station (GCS) software, which runs on a PC. The GCS is used to set up the drone, visualize the telemetry data sent by the drone during flight, and plan and execute autonomous missions. The GCS can also be used to control ArduPilot drones in real-time. By connecting a USB joystick to the PC running the GCS, pilots can manually control the drone. The joystick control values are sent over a telemetry radio link to the drone. The most widely used opensource GCS is Mission Planner which is shown in Figure 3.2.

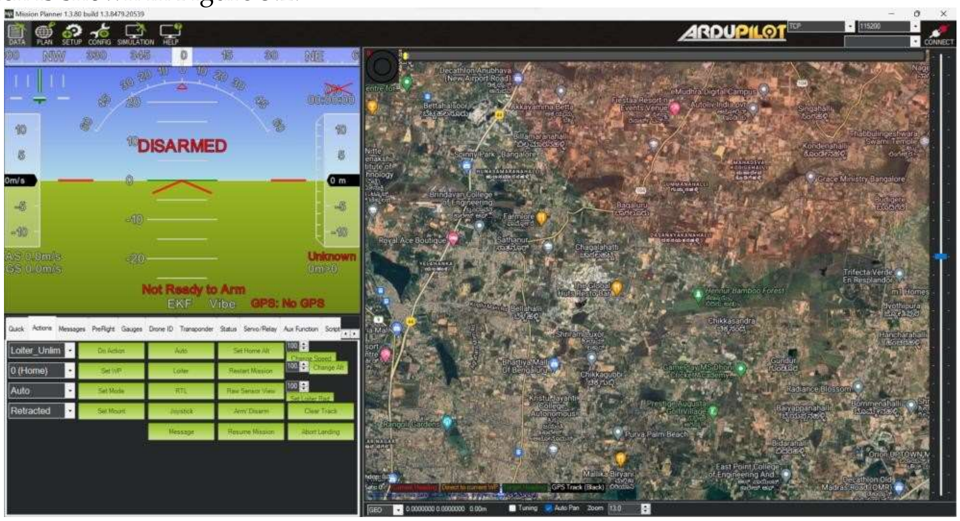


Figure 3.2: The Mission Planner Application.

1.2 Working Principle

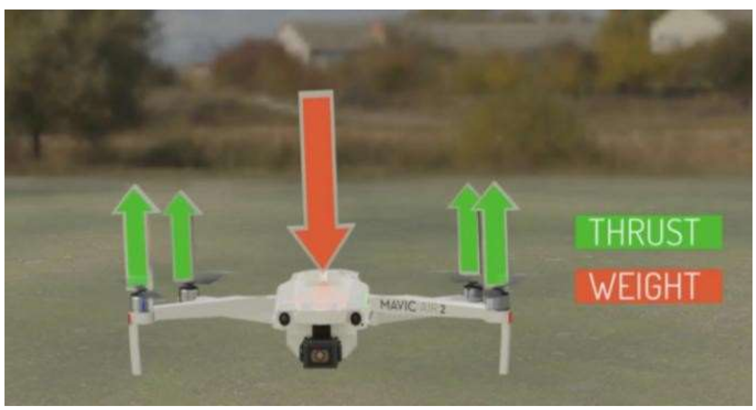


Figure 3.3: Forces acting on a drone while hovering stationarily.

The four primary forces acting on a drone while hovering stationarily are lift (L), gravity (G), thrust (T), and drag (D), each playing a crucial role in the flight dynamics of the drone as shown in figure 3.3.

Lift (L) is the upward force that is generated by the drone's propellers. The amount of lift generated depends on the speed of the propellers (ω), their size (A), and their pitch (α). This is shown using equation 3.1.

$$◎ = ◎_{◎} \; ◎ \; ◎^{2} \; ◎ \; ◎ \; ◎ \; ◎ \quad (3.1)$$

where $◎_{◎}$ is a constant that depends on the air density and the shape of the propellers.

Gravity (G) is the downward force exerted on the drone by the Earth. It is calculated as:

$$◎ = ◎ \; ◎ \; ◎$$

where m is the mass of the drone and g is the acceleration due to gravity.

Thrust (T) is the forward force that propels the drone. It is generated by tilting the drone, which redirects some of the lift force horizontally. The thrust can be represented as:

$$◎ = ◎ \; ◎ ◎◎◎(◎)$$

where θ is the tilt angle.

Drag (D) is the air resistance that opposes the motion of the drone. The amount of drag depends on the speed (v) and size of the drone (A), as well as the shape of the drone and its components. It can be represented as:

$$◎ = 0.5 \; ◎ \; ◎ \; ◎ \; ◎^{2} \; ◎ \; ◎_{◎} \; ◎ \; ◎ \qquad (3.2)$$

where ◎ is the air density, ◎ is the velocity of the drone, $◎_{◎}$ is the drag coefficient, and ◎ is the cross-sectional area of the drone.

The transformation matrix for a drone in 3D space can be represented by the

$$\begin{bmatrix} \cos(\theta)\cos(\psi) & -\cos(\varphi)\sin(\psi) + \sin(\varphi)\sin(\theta)\cos(\psi) & \sin(\varphi)\sin(\psi) + \cos(\varphi)\sin(\theta)\cos(\psi) \\ \cos(\theta)\sin(\psi) & \cos(\varphi)\cos(\psi) + \sin(\varphi)\sin(\theta)\sin(\psi) & -\sin(\varphi)\cos(\psi) + \cos(\varphi)\sin(\theta)\sin(\psi) \\ -\sin(\theta) & \sin(\varphi)\cos(\theta) & \cos(\varphi)\cos(\theta) \end{bmatrix}$$

, where ◎, ◎, and ◎ are the roll, pitch, and yaw angles respectively. This matrix is used to transform the drone's local coordinates to global coordinates.

1.3 Pitch

Figure 3.4: Forces acting on a drone during pitch motion.

The pitch motion in a drone is achieved by manipulating the speed of the rear motors. When the speed of the rear motors is increased, it generates a greater lift force at the back end of the drone compared to the front end. This differential in lift forces,

$$◎◎ = ◎_{◎◎◎◎} − ◎_{◎◎◎◎◎}$$

causes the drone to tilt forward, creating a pitch motion.

As the drone tilts forward, the thrust vector, which was initially pointing straight up opposing gravity, now has a component in the forward direction. This forward component of the thrust vector,

$$◎_{◎◎◎◎◎◎◎} = ◎_{◎◎◎◎◎◎} ◎ \sin(◎)$$

where ◎ is the tilt angle, generates a resultant force that propels the drone forward.

It's important to note that while the drone is pitching forward, the total lift force must still balance out the weight of the drone to keep it at a constant altitude. This is achieved by increasing the overall thrust (by speeding up all motors) while pitching. The vertical component of the thrust,

$$◎_{vertical} = ◎_{thrust} ◎ \cos(◎)$$

must equal the weight of the drone,

$$◎_{◎◎◎◎◎◎◎} = ◎. ◎$$

where ◎ is the mass of the drone and ◎ is the acceleration due to gravity. The precise control of motor speeds for maintaining altitude while achieving the desired pitch motion is handled by the drone's flight controller.

1.3.1 Roll

Figure 3.5. Forces acting on a drone during roll motion.

The roll motion in a drone is similar to the pitch motion, but it involves the manipulation of the left and right motors instead of the front and rear ones.

When the speed of the right motors is increased, it generates a greater lift force on the right side of the drone compared to the left side. This differential in lift forces,

$$◎◎ = ◎_{◎◎◎◎◎} − ◎_{◎◎◎◎}$$

causes the drone to tilt to the left, creating a roll motion. Conversely, increasing the speed of the left motors will cause the drone to roll to the right.

As the drone tilts, the direction of the thrust vector changes. The vertical component of the thrust,

$$◎_{vertical} = ◎_{thrust} ◎ \cos(◎)$$

continues to oppose gravity and maintain altitude, where ◎ is the tilt angle. The horizontal component of the thrust,

$$◎_{horizontal} = ◎_{thrust} ◎ \sin(◎)$$

Creates a sideways force that moves the drone in the direction of the roll.

Here,

$◎_{thrust}$ is the total thrust force.

While the drone is rolling, the total lift force must still balance out the weight of the drone to keep it at a constant altitude. This is achieved by increasing the overall thrust (by speeding up all motors) while rolling. The precise control of motor speeds for maintaining altitude while achieving the desired roll motion is handled by the drone's flight controller.

1.3.2 Yaw

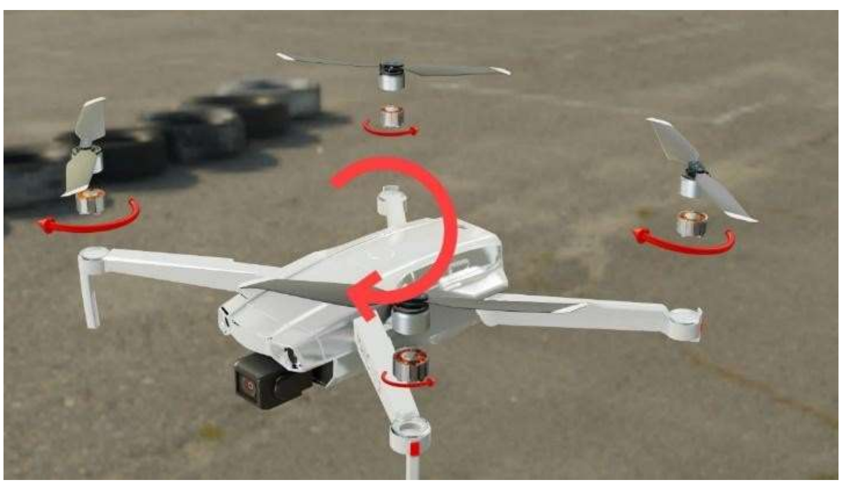


Figure 3.6: Torque acting on a drone during yaw motion.

The yaw motion in a drone is achieved by creating a net torque (τ) around the drone's centre of gravity as shown in figure 3.6. This is done by varying the speeds of motors spinning in opposite directions. The net torque can be calculated by

$$◎_{◎◎◎} = ◎1 + ◎2 − ◎3 − ◎4$$

Here, $◎_1$ and $◎_2$ are the torques generated by the motors spinning counterclockwise, and $◎_3$ and $◎_4$ are the torques generated by the motors spinning clockwise. A positive net torque indicates a clockwise rotation, and a negative net torque indicates a counterclockwise rotation.

During yaw motion, the total lift force (F) remains constant to maintain the drone's altitude. The total lift force can be calculated as:

$$◎_{◎◎◎} = ◎_1 + ◎_2 + ◎_3 + ◎_4$$

Even though the speeds of the motors are varied, the total lift force should remain constant. This means that any increase in force in one motor should be compensated by a decrease in force in the opposite motor. The drone's flight controller manages these adjustments to ensure stable flight.

Results

Performance Parameters

Comprehending and measuring critical performance metrics is essential for the thorough assessment of pilot performance in the context of drone simulation across five distinct levels of complexity. This subheading explores the complexities of evaluating pilot proficiency by closely examining three key metrics:

1. The accuracy of manoeuvring through hoops
2. The efficiency demonstrated by the completion time
3. The time taken to complete these manoeuvres

Testing Rubrics

Our assessment framework includes three important metrics: Hoops Missed, Collided Hoops, and Time Taken. This scoring scheme is designed to reward accuracy, speed, and fairness in the completion of the task, guaranteeing a thorough and impartial evaluation.

Table 4.1. Penalty System for Hoops

Penalty On Missing Hoops	Penalty On Collision
2	1

Table 4.2. Penalty System for Time Taken

Time Taken	Points Deducted
0 – 2 mins	0
2 - 3 mins	10
3 – 4 mins	20
4 – 5 mins	30
5 - 6 mins	40
Above 6 mins	50

Formula:

$$\text{Score}=100-(\alpha+\beta+\gamma)$$

Where:

- α represents the Penalty on Missing Hoops, calculated as the Number of Missed Hoops × Penalty Per Missed Hoop.
- β represents the Penalty on Collision, calculated as Number of Collided Hoops × Penalty Per Collided Hoop.
- γ represents Time Taken Points, with deductions based on the predefined time brackets

Novice 1 Before Training:

Table 4.3. Novice 1 Pilot's Performance Before Training

Levels	Total Number of Hoops	Hoops Missed	Collided Hoops	Time Taken	Score
Level 5	15	4	8	317s	48

Novice 1 After Training:

Table 4.4. Novice 1 Pilot's Performance After Training

Levels	Total Number of Hoops	Hoops Missed	Collided Hoops	Time Taken	Score
Level 1	10	3	2	283s	62
Level 2	10	2	2	275s	64
Level 3	12	3	3	277s	61
Level 4	12	2	1	267s	65
Level 5	15	1	2	259s	66

Total Score = 63.6
Performance improvement = 22.31%
Novice 2 Before Training:

Table 4.5. Novice 2 Pilot's Performance Before Training

Levels	Total Number of Hoops	Hoops Missed	Collided Hoops	Time Taken	Score
Level 5	15	3	5	308s	49

Novice 2 After Training:

Table 4.6. Novice 2 Pilot's Performance After Training

Levels	Total Number of Hoops	Hoops Missed	Collided Hoops	Time Taken	Score
Level 1	10	3	2	309s	52
Level 2	10	1	1	292s	67
Level 3	12	2	3	301s	53
Level 4	12	1	1	289s	67
Level 5	15	1	2	294s	66

Total Score = 61
Performance improvement = 24.49%
Novice 3 Before Training:

Table 4.7. Novice 3 Pilot's Performance Before Training

Levels	Total Number of Hoops	Hoops Missed	Collided Hoops	Time Taken	Score
Level 5	15	2	8	310s	48

Novice 3 After Training:

Table 4.8. Novice 3 Pilot's Performance After Training

Levels	Total Number of Hoops	Hoops Missed	Collided Hoops	Time Taken	Score
Level 1	10	3	3	302s	51
Level 2	10	1	2	291s	66
Level 3	12	2	4	315s	52
Level 4	12	2	2	287s	64
Level 5	15	2	3	278s	63

Total Score = 59.2
Performance improvement = 23.33 %

Novice 4 Before Training:

Table 4.9. Novice 4 Pilot's Performance Before Training

Levels	Total Number of Hoops	Hoops Missed	Collided Hoops	Time Taken	Score
Level 5	15	4	8	325s	44

Novice 4 After Training:

Table 4.10 Novice 4 Pilot's Performance After Training

Levels	Total Number of Hoops	Hoops Missed	Collided Hoops	Time Taken	Score
Level 1	10	2	4	314s	52
Level 2	10	3	3	302s	51
Level 3	12	2	5	320s	51
Level 4	12	2	6	296s	60
Level 5	15	2	4	280s	62

Total Score = 55.2
Performance improvement = 25.45%

Novice 5 Before Training:

Table 4.11. Novice 5 Pilot's Performance Before Training

Levels	Total Number of Hoops	Hoops Missed	Collided Hoops	Time Taken	Score
Level 5	15	5	9	346s	41

Novice 5 After Training:

Table 4.12. Novice 5 Pilot's Performance After Training

Levels	Total Number of Hoops	Hoops Missed	Collided Hoops	Time Taken	Score
Level 1	10	4	8	315s	44
Level 2	10	2	5	325s	51
Level 3	12	2	6	340s	50
Level 4	12	3	4	297s	60

Level 5	15	1	3	310s	55

Total Score = 52
Performance improvement = 26.83%

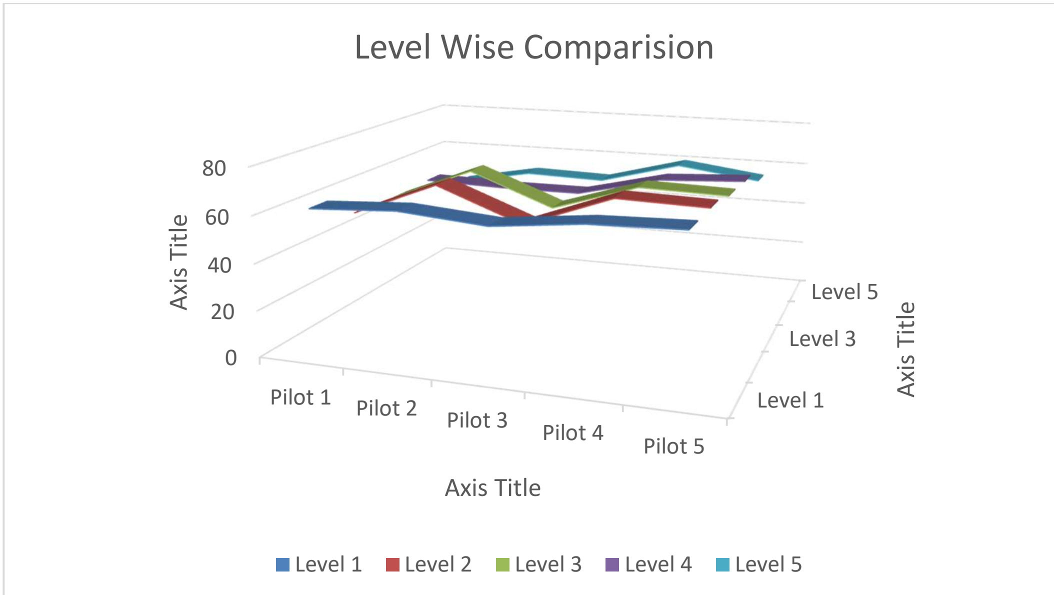


Figure 4.1. Three-Dimensional Level Wise Comparison

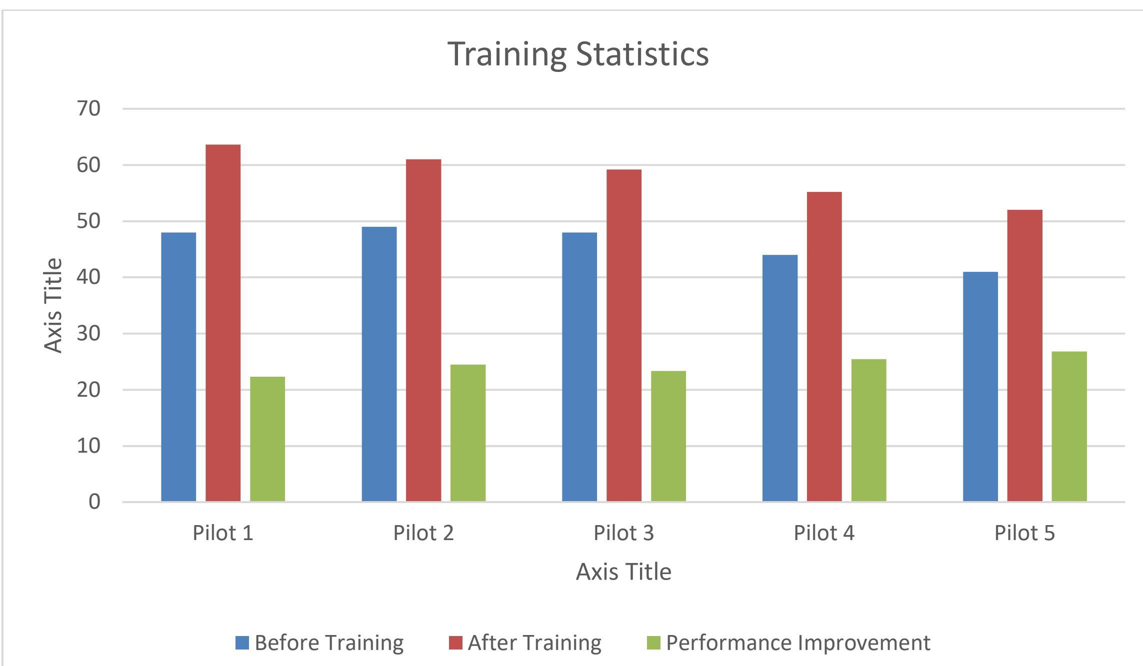


Figure 4.2. Training Statistics of Novice Pilots

Comparative Analysis of Novice and Experienced Drone Pilots

The comparative analysis of the scoring formulas between inexperienced and experienced drone pilots provides a thorough understanding of the increasing levels of competence and proficiency attained when operating in virtual environments.

To fully evaluate drone pilot competency, we carried out a detailed investigation with both inexperienced and experienced pilots flying our simulator courses. Five inexperienced pilots who are still in the training phase were compared to five seasoned professionals who have proven to have advanced drone piloting abilities.

Experienced Pilots:

Table 4.13. Experienced Pilot's Performance

Experienced Pilots	**Score**
Pilot 1	66.37
Pilot 2	68.43
Pilot 3	69.55
Pilot 4	70.01
Pilot 5	63.79

Average Score =71.488

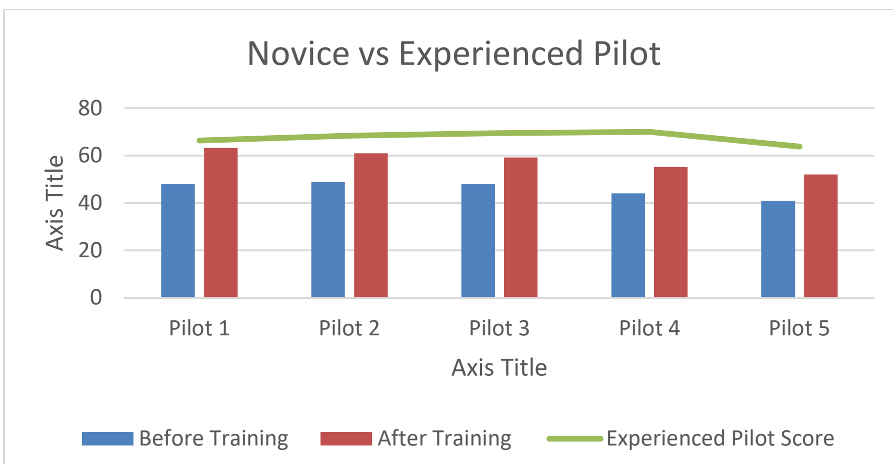


Figure 4.3. Comparison of Novice vs Experienced Pilot

Novice Pilots:

Table 4.14. Novice Pilot's Performance

Novice Pilots	Before Training	After Training	Performance Improvement
Pilot 1	48	63.3	22.31%
Pilot 2	49	61	24.49%
Pilot 3	48	59.2	23.33%
Pilot 4	44	55.2	25.4%
Pilot 5	41	52	26.83%

Average Score Before Training = 46
Average Score After Training = 67.63
Average Performance of Performance Improvement = 24.47%

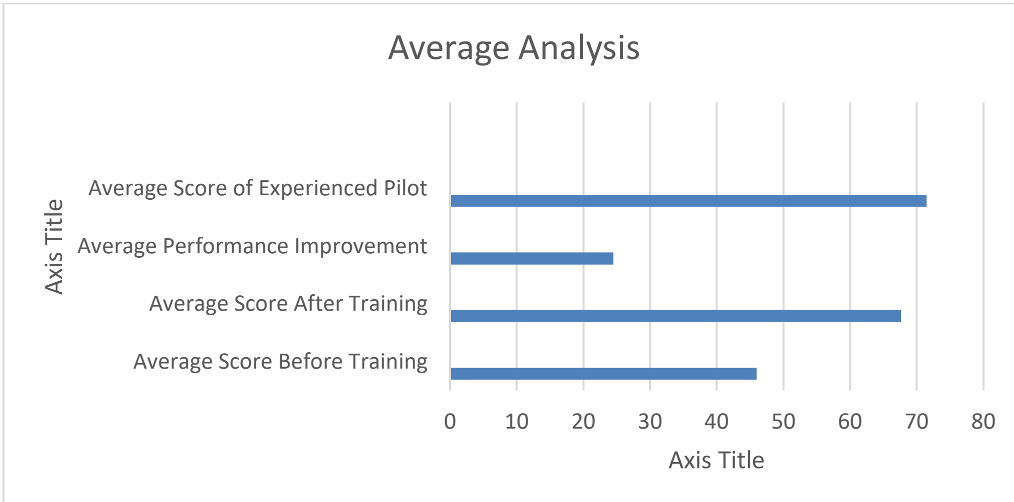


Figure 4.4. Analysis of Average Metrics

Conclusion

In the current setup, the passthrough boundary is statically set to the boundary of the cockpit. This requires a setup where the headset is placed on a designated dock on the cockpit to adjust the boundary when starting the application, effectively tailoring the application to the cockpit. However, a more flexible solution could involve the headset tracking the cockpit using computer vision with markers or ideally active trackers. Unfortunately, the image quality of the cameras of the headset are not currently suitable for performing computer vision algorithms, and there are no active markers that work with the Meta Quest Pro.

The parameters used in this application concerning the throttle curve, agility, etc., are left at their default settings to allow for wider acceptability to novice pilots. However, experienced pilots may prefer different curves and different modes of operation. This suggests a need for a setup and adjustment of parameters to tailor the simulator to different pilots, enhancing its usability and adaptability.The physics backend could benefit from using a description format like the Unified Robot Description Format (URDF). URDF is an XML-based file format used in the field of robotics to describe the structure and kinematics of a robot. This would allow for greater interoperability and usage with different types of vehicles. The cockpit could also be augmented with virtual gauges and displays that show information regarding the current flight such as altitude, velocity, and heading. This is currently directly augmented onto the virtual screen with the view of the drone.

References

1) Ahmed A. Moustafa, Srinivasa Chakravarthy, Joseph R. Phillips, AnkurGupta, Szabolcs Keri, Bertalan Polner, Michael J. Frank, and Marjan Jahanshahi. 2016. Motor symptoms in Parkinson's disease: A unified framework. Neuroscience and Biobehavioral Reviews 68 (Sept. 2016), 727–740. doi:10.1016/j.neubiorev.2016.07.

2) Michael Ora Powell, Aviv Elor, Mircea Teodorescu, and Sri Kurniawan. 2020. OpenButterfly: Multimodal Rehabilitation Analysis of Immersive Virtual Reality for Physical Therapy. American Journal of Sports Science and Medicine 8, 1 (June 2020), 23–35. doi:10.12691/ajssm-8-1-5 Number: 1 Publisher: Science and

Education Publishing.

3) Clifford A. Reilly, Aimee Burnett Greeley, David S. Jevsevar, and Ida Leah Gitajn. 2021. Virtual reality-based physical therapy for patients with lower extremity injuries: feasibility and acceptability. OTA International 4, 2 (May 2021), e132. doi:10.1097/OI9.0000000000000132

4) Gregory N. Ruegsegger and Frank W. Booth. 2018. Health Benefits of Exercise. Cold Spring Harbor Perspectives in Medicine 8, 7 (July 2018), a029694. doi:10.1101/ cshperspect.a029694

5) Afrooz Seyedebrahimi, Reza Khosrowabadi, and Hossein Mousavi Hondori. 2019. Brain Mechanism in the Human-Computer Interaction Modes Leading to Different Motor Performance. doi:10.1109/IranianCEE.2019.8786750 Pages: 1806.

6) Jawaria Shahid, Ayesha Kashif, and Muhammad Kashif Shahid. 2023. A Com prehensive Review of Physical Therapy Interventions for Stroke Rehabilitation: Impairment-Based Approaches and Functional Goals. Brain Sciences 13, 5 (April 2023), 717. doi:10.3390/brainsci13050717

7) Dan Shao and I-Jui Lee. 2020. Acceptance and Influencing Factors of Social Virtual Reality in the Urban Elderly. Sustainability 12, 22 (Nov. 2020), 9345. doi:10.3390/su12229345

8) William R. Sherman and Alan B. Craig. 2018. Understanding Virtual Reality: Interface, Application, and Design. Morgan Kaufmann.

9) Arthur A. Stone, Saul Shiffman, Joseph E. Schwartz, Joan E. Broderick, and Michael R. Hufford. 2002. Patient non-compliance with paper diaries. BMJ (Clinical research ed.) 324, 7347 (May 2002), 1193–1194. doi:10.1136/bmj.324.7347.

10) Richard Tang, Xing-Dong Yang, Scott Bateman, Joaquim Jorge, and Anthony Tang. 2015. Physio@Home: Exploring Visual Guidance and Feedback Techniques for Physiotherapy Exercises. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM, Seoul Republic of Korea, 4123–4132. doi:10.1145/2702123.2702401

11) Yuan Tian, Suraj Raghuraman, Thiru Annaswamy, Aleksander Borresen, Klara Nahrstedt, and Balakrishnan Prabhakaran. 2017. H-TIME: Haptic-enabled Tele Immersive Musculoskeletal Examination. In Proceedings of the 25th ACM interna tional conference on Multimedia (MM '17). Association for Computing Machinery, NewYork, NY, USA, 137–145. doi:10.1145/3123266.3123395.

12) Celine Timmermans, MelvynRoerdink, Marielle W. van Ooijen, Carel G. Meskers, Thomas W. Janssen, and Peter J. Beek. 2016. Walking adaptability therapy after stroke: study protocol for a randomized controlled trial. Trials 17, 1 (Aug. 2016), 425. doi:10.1186/s13063-016-1527-6.

13) João Vieira, Mauricio Sousa, Artur Miguel Arsénio, and Joaquim Jorge. 2015. Augmented Reality for Rehabilitation Using Multimodal Feedback. doi:10.1145/ 2838944.2838954.

14) Matt Whitlock, Stephen Smart, and Danielle Albers Szafir. 2020. Graphical Perception for Immersive Analytics. In 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). 616–625. doi:10.1109/VR46266.2020.00084 ISSN: 2642-5254.

15) Hemendra Worlikar, Sean Coleman, Jack Kelly, Sadhbh O'Connor, Aoife Murray, Terri McVeigh, Jennifer Doran, Ian McCabe, and Derek O'Keeffe. 2023. Mixed Reality Platforms in Telehealth Delivery: Scoping Review. JMIR Biomedical Engineering 8 (March 2023), e42709. doi:10.2196/42709.

16) Dr Prakash Kuppuswamy, Ms. Vijaya Ramineni, Dr Suhas G K, Dr Nagendra Kumar M, Mounika Samudrapu, Preeti Nashier, "A Comprehensive Study on the Performance and Security of Post Quantum Cryptography (PQC) Compared to Classical RSA Algorithm", International Conference on Web Intelligence and Human-Machine Interaction (ICWIHI 2026) Springer conference, April 24-25, 2026. https://icwihi.com/2026/

17) Purushottam Kumar, Dolly Sharma, Samkit Jain, Prakash Kuppuswamy, Current flow centrality based link prediction in unipartite and bipartite complex networks, *Journal of Complex Networks*, Volume 14, Issue 2, April 2026, cnag010, https://doi.org/10.1093/comnet/cnag010 (Q2-KL-affiliation)