A Review on Mixed Reality-Based Simulator for Drone Pilot Training

Suhas G K[1], Shashi Kant Gupta[2]

[1]Post Doctoral research scholar, Lincoln University college, Malaysia

[2]Professor, Lincoln University college, Malaysia

[1]suhask300@gmail.com

Abstract: In recent years, drones have grown in popularity due to their capacity to carry out various activities faster and more successfully than conventional techniques. Drones are employed in a variety of industries, including photography, search and rescue, surveying, and agriculture. However, due to the sophisticated nature of the technology, operating a drone requires specialized knowledge and training. Training drone pilots can be difficult and expensive since it requires hands-on experience and regulatory and safety measures expertise. The use of simulators is one potential answer to the difficulties in training drone pilots. Simulators offer operators a safe, controlled environment where they may hone their abilities and acquire experience without having to worry about endangering their equipment or hurting others. The challenge at hand involves developing an immersive drone flight training system that seamlessly integrates with the real world, ensuring users experience a heightened sense of engagement without motion sickness. The simulation must employ physics that precisely mirror real drone flight dynamics, eschewing fictional models for authenticity. Central to this objective is the implementation of a user interface providing accurate telemetry values, affording pilots a comprehensive understanding of crucial realistic factors during flight. Additionally, the system demands a sophisticated training module that imparts practical skills and knowledge, ensuring proficiency in drone operation through real-world scenarios. The limitations of simulators, such as their resemblance to VR games rather than serious training environments, pose challenges for training drone pilots. Our research focuses on addressing these challenges by closely modelling the virtual environment to mirror the real environment using tools like Unity3D and Blender. Additionally, efforts have been made to model the physics of the drone by applying static and dynamic propeller thrust based on RPM data from the open-source flight planner ArduPilot. This work also aims to provide users with control by enabling them to switch between various views, including first person, third person, and fixed views. The testing procedure employed a hoop course system. This involved a skilled drone pilot navigating through progressively challenging courses, with factors such as reduced hoop size, intricate manoeuvres, increased hoop quantity, and diminished lighting contributing to heightened difficulty levels. The pilot's score and time served as the benchmark. Novice pilots then tested the application, with their scores and times recorded. As they approached the benchmark, course difficulty increased.

Keywords: Augmented Reality, Virtual Reality, Mixed Reality, ArduPilot

1 Introduction

Motivated by the need to provide safe, inexpensive, and enjoyable environments for flying real drones, this chapter introduces a Mixed Reality drone simulator. This initiative aims to understand the broader context and associated terminologies of this endeavour. Introduction to Mixed Reality Virtual Reality (VR) and Augmented Reality (AR) have significantly impacted how users engage with computer-generated environments. While VR immerses users in virtual spaces, AR integrates digital content into the real world. However, the noticeable separation between real and virtual

realms presents a problem for AR and reduces user immersion. [5].Mixed Reality (MR) addresses this challenge by seamlessly merging the real and virtual worlds. It describes a continuum in which a person's perception of the real world is blended with computer-generated content [4]. This is accomplished through a dynamic interface that allows virtual and real-world objects to interact. [6]. This allows a dynamic and immersive user experience [5] while still allowing the user to interact normally with his surroundings. 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].

1.1 Overview Diagram

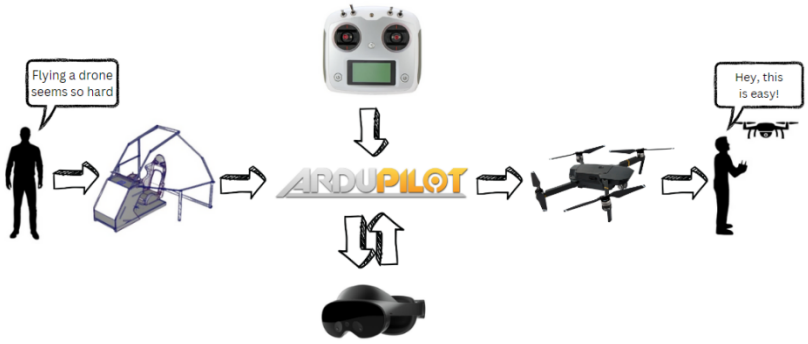


Fig 1.1: Overview Diagram

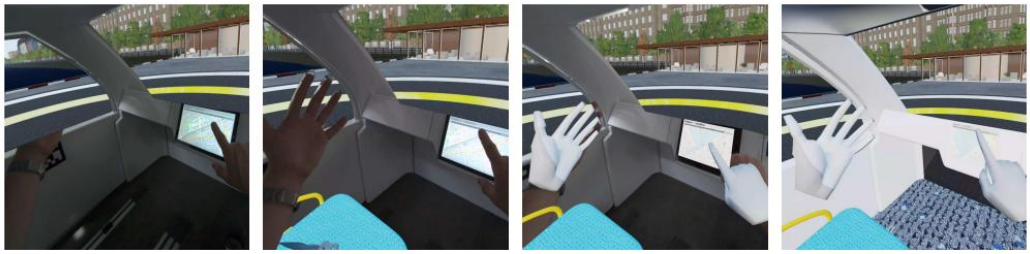
The objective of the project, as depicted in Figure 1.1, is to facilitate drone piloting training for novices through a dedicated drone cockpit simulation, allowing them to be able to fly drones with ease after the training is complete. Within this simulated cockpit, a person is equipped with a controller and a headset. The controller transmits user input values to the ArduPilot system, where essential computations take place. Subsequently, ArduPilot communicates the computed revolutions per minute (RPM) values to the Meta Quest Pro headset. These RPM values are crucial for controlling and navigating the drone within the virtual environment.

Effectively, the user, while seated inside the cockpit, undergoes drone simulation training, with the Meta Quest Pro translating the RPM data into actionable flight commands. Simultaneously, telemetry values and coordinate data generated during the simulation are conveyed back to the ArduPilot system. This integrated system ensures a seamless and comprehensive training experience, empowering novices to acquire the skills necessary for flying drones proficiently.

2 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: 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].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].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].

The cockpit dashboard (Cessna 152) [8]

3 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.

3.1 Architecture

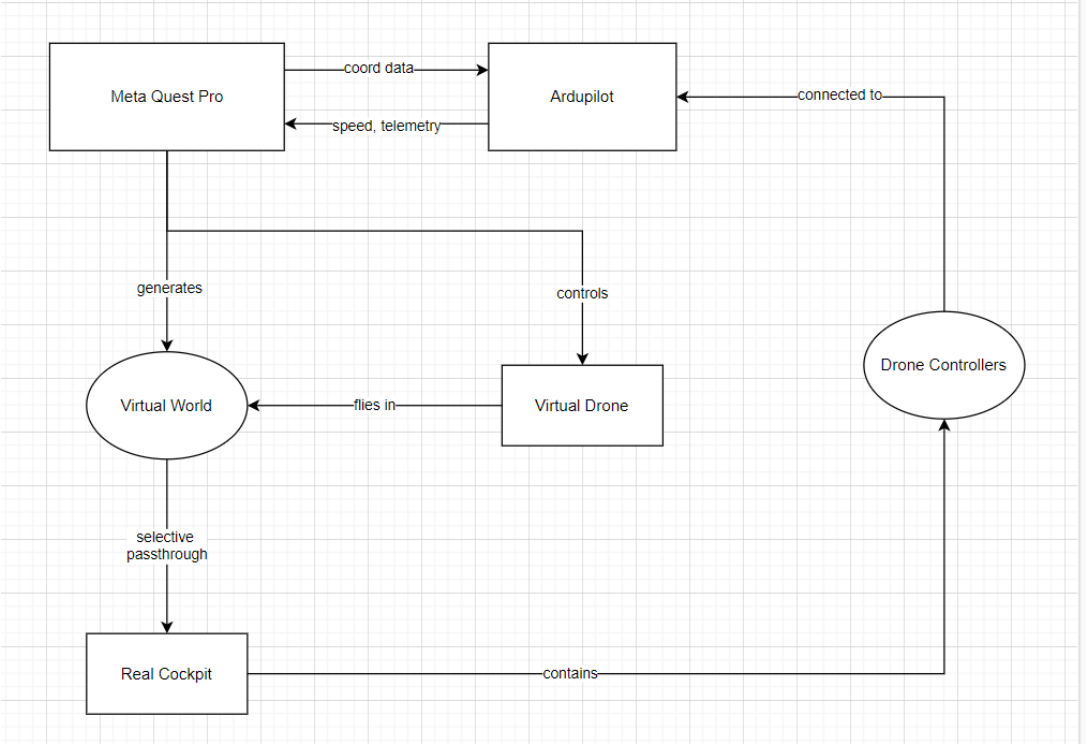


Figure 3.1: Architecture of the proposed system

As depicted in the Figure 3.1, in this setup, the user is seated inside a mock cockpit equipped with a drone joystick controller. This controller is connected to the ArduPilot Software in the Loop (SITL) running inside the Mission Planner Application via a USB connection. The joystick controls are mapped to the RC channels of Mission Planner, which continuously polls data from the joystick.The ArduPilot SITL establishes a wireless UDP connection with a Unity application running on the Meta Quest Pro headset. This application features a virtual world where a virtual drone is flown. It also provides a real-world passthrough of the cockpit, which is displayed inside the headset.The Unity application receives a JSON object from the ArduPilot SITL. This object contains the calculated Pulse Width Modulation (PWM) values for all the motors. The application applies these values to the four motors of the virtual drone, causing it to hover or move in a certain direction.In return, the Unity application sends back another JSON object to the ArduPilot SITL. This object contains various parameters of the virtual drone in flight, including its position, velocity, acceleration, etc. This exchange of data between the ArduPilot SITL and the Unity application is performed continuously, allowing for real-time control and feedback of the virtual drone's flight.

3.2 Configuration of simulation environment

3.2.1 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.

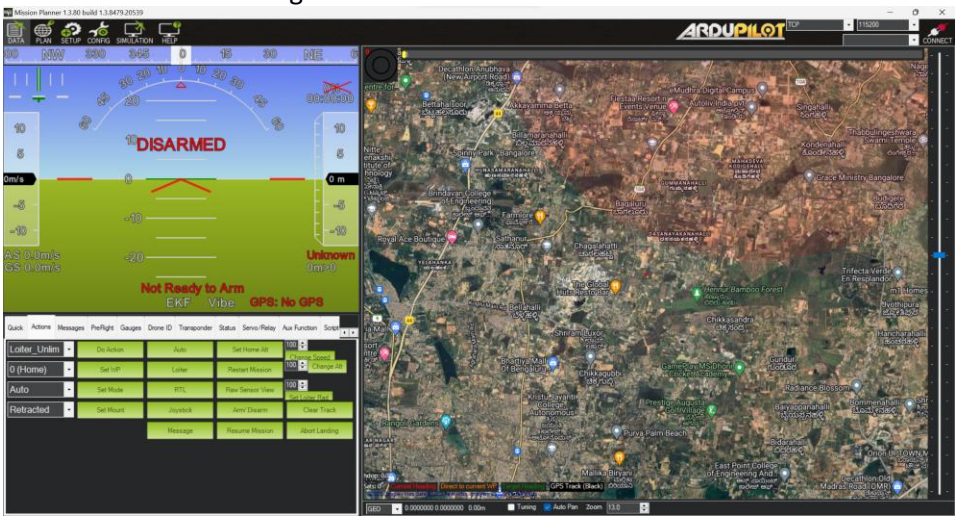


Figure 3.2: The Mission Planner Application.

3.3 Working Principle

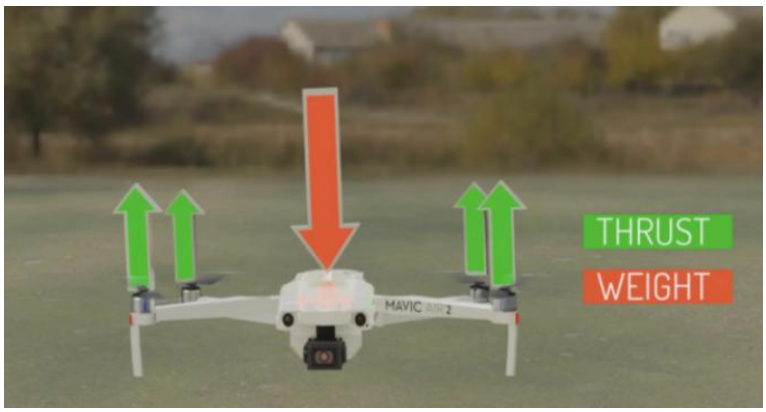


Figure 3.: Forces acting on a drone while hovering stationarily.

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.xxx.

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.

$$L = k_L * \omega^2 * A * \alpha \quad (3.1)$$

where k_L 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:

$$G = m * g$$

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:

$$T = L \cdot sin(\theta)$$

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:

$$D = 0.5 * \rho * v^2 * C_D * A \quad (3.2)$$

where ρ is the air density, v is the velocity of the drone, C_D is the drag coefficient, and A 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 φ, θ, and ψ are the roll, pitch, and yaw angles respectively. This matrix is used to transform the drone's local coordinates to global coordinates.

3.3.1 Pitch

Figure 3.xx: 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,

$$\Delta F = F_{rear} - F_{front}$$

, 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,

$$F_{forward} = F_{thrust} \cdot \sin(\theta)$$

, where θ 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,

$$F_{\text{vertical}} = F_{\text{thrust}} \cdot \cos(\theta)$$

, must equal the weight of the drone,

$$F_{gravity} = m.g$$

where m is the mass of the drone and g 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.

3.3.2 Roll

Figure 3.11: 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,

$$\Delta F = F_{right} - F_{left}$$

, 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,

$$F_{\text{vertical}} = F_{\text{thrust}} \cdot \cos(\theta)$$

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

$$F_{\text{horizontal}} = F_{\text{thrust}} \cdot \sin(\theta)$$

, creates a sideways force that moves the drone in the direction of the roll. Here, F_{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.

3.3.3 Yaw

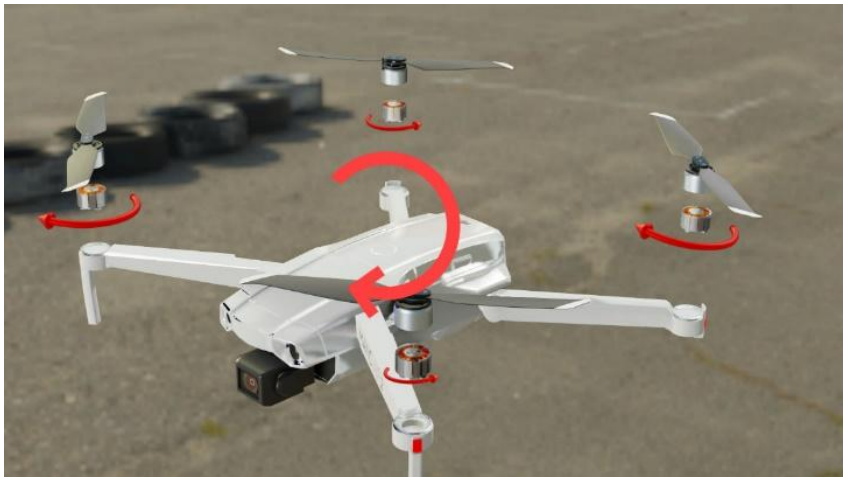


Figure 3.xx: Torque acting on a drone during yaw motion.

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. xx. This is done by varying the speeds of motors spinning in opposite directions. The net torque can be calculated by

$$\tau_{net} = \tau_1 + \tau_2 - \tau_3 - \tau_4$$

Here, $\tau_1$and τ_2 are the torques generated by the motors spinning counterclockwise, and τ_3 and τ_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:

$$F_{net} = F_1 + F_2 + F_3 + F_4$$

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.

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, Ankur Gupta, 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.2016.07.010

[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.324.7347.1193

[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