IMMERSIVE MULTISENSORY VR MOTION SYSTEM FOR FULL BODY NAVIGATION

INDIVIDUAL THESIS

Gunawardena L. H

(IT22082992)

Individual component: Motion Rendering and Control for Embodied VR Navigation

Group ID: R26-IM-004

Report submitted in partial fulfilment of the requirements for the
Bachelor of Science (Hons) in Information Technology

Specializing in Interactive Media

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

October 2026

DECLARATION AND SUPERVISOR CERTIFICATION

Candidate declaration. I declare that this thesis is my own work and does not incorporate, without acknowledgement, material previously submitted for a degree or diploma at any university or institute of higher learning. To the best of my knowledge, it contains no material previously published or written by another person except where acknowledgement is made in the text. I grant the Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute this thesis, in whole or in part, in print, electronic, or other media, while retaining the right to use the content in future work.

Academic integrity and AI disclosure. I have disclosed all use of generative AI and other automated tools in accordance with the module policy. I remain responsible for the accuracy, originality, citations, code, data, analyses, and conclusions in this thesis.

Student name: Gunawardena L H

Student ID: IT22082992

Signature: _____ Date: _____

Supervisor certification. The above candidate carried out this research for the bachelor's degree thesis under my supervision.

Role and name	Signature	Date
Supervisor: Mr. Ishara Gamage		
Co-supervisor: Mr. Jeewaka Perera		

ABSTRACT

Most virtual reality (VR) flying experiences rely on handheld joysticks. This separation between what a user sees visually and what their body physically feels often causes motion sickness, dizziness, and breaks overall immersion. While physical motion platforms help bridge this gap, translating continuous, agile aerial movement onto a compact motion seat is challenging because mechanical actuators have strict tilt limits and can easily bottom out.

This research develops a real-time motion rendering and flight control framework designed for a broom-based, seated VR navigation platform. Following a Design Science Research methodology, an end-to-end control system was created to deliver intuitive full-body navigation. Forward speed and vertical altitude are controlled naturally by squeezing a handle-mounted Force Sensing Resistor (FSR), while directional turns are captured through head-mounted gyroscope tracking. Within Unity, a custom motion rendering engine extracts flight data—including position, velocity, and acceleration—and uses linear interpolation (Mathf.Lerp) smoothing alongside dead zone filtering to prevent unintended steering drift and jerky movement. These computed flight commands are transmitted through a serial communication interface to a microcontroller that drives the physical seat actuators within safe mechanical thresholds.

The system was evaluated through user trials involving 35 participants navigating a flight course. The input mechanics proved highly responsive, with 74.3% of participants rating the pressure-based throttle and altitude response positively, achieving an average rating of 4.2 out of 5. The evaluation also identified key areas for improvement, such as physical grip fatigue over extended sessions and rotational drift during sharp head turns. This individual component contributes the physical grip input pipeline, the motion filtering and rendering algorithms, the embedded actuator control link, and the empirical user validation.

Keywords: VR Locomotion, Motion Rendering, Actuator Control, Force Sensing Resistor, Embodied Interaction

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, co-supervisor, external supervisors and the academic staff of the Faculty of Computing at the Sri Lanka Institute of Information Technology (SLIIT) for their continuous guidance, insightful feedback, and encouragement throughout the design and development of this research project. Their advice was invaluable in refining the motion cueing strategies, control algorithms, and safety parameters of this work.

I am grateful to my team members of Project Group R26-IM-004 for their seamless collaboration, dedication, and technical synergy in constructing and integrating the complete multisensory VR motion platform.

Finally, I extend my heartfelt thanks to the 35 participants who volunteered their time to take part in the experimental trials and user evaluations. Their constructive feedback and engagement made the empirical validation of this flight motion system possible.

TABLE OF CONTENTS

1 INTRODUCTION 1
1.1 Overall project context 1
1.2 Individual component context 2
1.3 Problem statement 3
1.4 Research gap 4
1.5 Aim and objectives 5
1.6 Research questions or hypotheses 6
1.7 Scope limitations and assumptions 6
1.8 Individual contribution 7
1.9 Thesis structure 9
2 LITERATURE REVIEW AND RELATED WORK 10
2.1 Review method and scope 10
2.2 Domain and theoretical foundations 10
2.2.1 Sensory Conflict Theory and Cybersickness 10
2.2.2 Vestibular Motion Perception and Tilt-Coordination 11
2.3 Related research 12
2.3.1 Embodied VR Locomotion Systems 12
2.3.2 Seated Motion Cueing Implementations 12
2.4 Existing systems products and techniques 13
2.5 Gap synthesis and conceptual framework 14
2.5.1 Gap Synthesis 14
2.5.2 Conceptual Framework 15
3 RESEARCH METHODOLOGY 16
3.1 Research design 16
3.2 Data participants or study materials 16
3.2.1 Participant Sample 16
3.2.2 Experimental Materials and Apparatus 17
3.3 Experimental or study procedure 17
3.4 Measures and evaluation metrics 18
3.4.1 System Performance Metrics 18
3.4.2 User Experience Metrics 18
3.5 Data analysis 18
3.6 Ethics privacy security and responsible AI 19

3.7 Reproducibility and research artefacts 20
4 PRODUCT AND SOLUTION DESIGN 21
4.1 Requirements 21
4.1.1 Functional Requirements 21
4.1.2 Non-Functional Requirements 22
4.2 Component architecture within the complete solution 22
4.3 Design alternatives and decisions 24
4.4 Detailed design 25
4.5 Standards and professional practice 25
5 IMPLEMENTATION INTEGRATION AND TESTING 26
5.1 Development environment and technology stack 26
5.2 Component implementation 26
5.3 Integration with the other components 27
5.4 Verification and software testing 27
5.5 Deployment and operation 28
6 RESULTS AND VALIDATION 29
6.1 Research results 29
6.2 Product and technical results 30
6.3 User or stakeholder validation 31
6.4 Integration results 31
6.5 Summary of findings 32
7 DISCUSSION 33
7.1 Interpretation of findings 33
7.2 Comparison with literature baselines and existing solutions 33
7.3 Research contribution 34
7.4 Product contribution 34
7.5 Limitations and threats to validity 35
7.6 Implications 36
8 CONCLUSION AND FUTURE DIRECTIONS 37
8.1 Conclusion 37
8.2 Achievement of objectives 37
8.3 Individual research and product contributions 38
8.3.1 Research Contributions 38
8.3.2 Product Contributions 38
8.4 Future directions 39

REFERENCES 40
GLOSSARY 42
APPENDICES 43
Appendix A Research Instruments or Extended Results 43
A.1 Post-Exposure System Evaluation Questionnaire 43
A.2 Extended User Response Distribution 44
A.3 Telemetry Data Frame Specification 44
Appendix B Academic Integrity and AI Use Disclosure 44
B.1 AI Tools and Usage Scope 44
B.2 Statement of Authorship and Verification 45

LIST OF FIGURES

Figure No.	Title	Page
Figure 1.1	Functional block architecture of the motion rendering and control subsystem (Original in Color)	9
Figure 2.1	Conceptual framework of the motion rendering and control pipeline (Original in Color)	16
Figure 3.1	Sequential experimental trial protocol for human-subject evaluation (Original in Color)	24

LIST OF TABLES

Table 1: Comparison of existing VR locomotion and motion rendering approaches.13
Table 2: Architectural design decisions and trade-off evaluations 24
Table 3: Participant evaluation of handle-grip throttle and altitude responsiveness. 31

LIST OF ABBREVIATIONS

Abbreviation	Meaning
2-DOF	Two Degrees of Freedom
3-DOF	Three Degrees of Freedom
6-DOF	Six Degrees of Freedom
AC	Alternating Current
ADC	Analog-to-Digital Converter
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
B2B	Business-to-Business
CAD	Computer-Aided Design
DC	Direct Current
DSR	Design Science Research
FEA	Finite Element Analysis
FoS	Factor of Safety
FSR	Force Sensing Resistor
HCI	Human-Computer Interaction
HMD	Head-Mounted Display
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
IP	Intellectual Property
ISO	International Organization for Standardization
IT	Information Technology
LBE	Location-Based Entertainment
LED	Light-Emitting Diode
MCU	Microcontroller Unit
MIG	Metal Inert Gas
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
NFR	Non-Functional Requirement
PWM	Pulse Width Modulation
SDG	Sustainable Development Goal
SHS	Square Hollow Section
SLIIT	Sri Lanka Institute of Information Technology
SSQ	Simulator Sickness Questionnaire
UART	Universal Asynchronous Receiver-Transmitter
UI	User Interface
USB	Universal Serial Bus
VR	Virtual Reality
WBS	Work Breakdown Structure
XOR	Exclusive OR

1 INTRODUCTION

1.1 Overall project context

Virtual reality (VR) technologies have advanced substantially over recent years, providing high-resolution head-mounted displays, wide viewing angles, and responsive optical tracking that render visual worlds with exceptional detail [1], [2]. Yet, despite these major visual improvements, the physical mechanics used to navigate through these virtual environments have remained mostly unchanged [4]. Most consumer VR applications continue to rely on traditional handheld controllers, analog thumbsticks, or artificial point-and-teleport mechanics [4], [10].

While thumbstick steering works adequately in conventional 2D screen games, it introduces severe interaction issues during fast-paced aerial simulations and three-dimensional flight [4], [7]. When a user wears a headset displaying rapid forward acceleration, steep vertical ascents, and sharp banked turns while their body remains stationary on an ordinary chair, a major sensory mismatch is created [1], [2]. The visual system registers rapid self-motion, known as vection, but the inner ear's vestibular organs and bodily proprioceptors report no movement at all [2], [9]. This sensory conflict is widely established as the primary cause of cybersickness, producing nausea, dizziness, disorientation, and rapid physical fatigue [2], [9], [10].

Furthermore, relying on abstract thumbsticks ruins the physical grounding and feeling of embodiment required for believable flight [1], [4]. In real life, piloting a vehicle or riding a craft involves physical resistance, balance shifts, and vestibular feedback [3], [4]. High-end industrial training simulators address this by placing cockpits on top of multi-axis hydraulic or pneumatic hexapod motion bases [3]. However, these industrial Stewart platforms are bulky, cost tens of thousands of dollars, and require specialized power infrastructure, making them impractical for portable setups, consumer entertainment, and localized arcade venues [3], [8]. Bridging the gap between affordable stationary seating and realistic, full-body vestibular motion remains an active engineering challenge [3], [4].

To solve this disconnect, the collaborative research project titled *Immersive Multisensory VR Motion System for Full Body Navigation* introduces a synchronized, embodied physical motion platform centered on a straddled flight apparatus. Instead of using thumbsticks, the user sits on a custom-fabricated, counterbalanced physical broomstick rig and steers by naturally shifting their posture and applying physical hand grip. The platform closes the sensory loop by combining four synchronized physical feedback channels in real time: a dynamic motion seat that provides pitch and roll vestibular cues, front and lateral airflow fans that scale wind output with virtual airspeed, a wearable vibrotactile haptic vest that renders atmospheric drag and localized turbulence, and external synchronized LED lighting that communicates in-game flight states to external spectators. By aligning physical sensations directly with what the user sees, the system provides a convincing, stable, and embodied flight experience [1], [5], [7].

1.2 Individual component context

Within the four-part multisensory platform, this individual thesis focuses directly on **Motion Rendering and Control for Embodied VR Navigation**. While environmental airflow simulation and vibrotactile vest actuators reproduce surface drag, ambient air currents, and localized turbulence [1], [5], [6], [7], physical motion of the seat serves as the fundamental driver for vestibular self-motion perception [1], [3]. The mechanical platform provides the onset vestibular cues that signal bodily acceleration, deceleration, and spatial banking [3].

The motion rendering and control component operates as the bidirectional software-and-hardware bridge situated between the virtual simulation running in Unity and the embedded electromechanical drive electronics. The functional scope of this component is split into two primary operational responsibilities:

1. **Embodied Flight Input Acquisition and Conditioning**: The subsystem continuously samples and conditions physical control inputs delivered by the user. This includes digitizing analog compression forces from a handle-

mounted Force Sensing Resistor (FSR) to modulate virtual forward throttle and vertical climb/descent rates, alongside real-time orientation tracking derived from the head-mounted display's internal inertial measurement unit (IMU) to govern horizontal yaw and roll banking maneuvers.

2. **Kinematic State Extraction and Motion Cueing**: The subsystem monitors the avatar's resulting 3D flight trajectory within Unity, extracts dynamic variables including spatial position, velocity vectors, and linear acceleration and translates these unconstrained states into physically viable, rate-limited tilt commands [3]. These commands are then streamed over a high-speed serial communication interface to a dedicated microcontroller that coordinates actuator motion under strict safety boundaries [3].

Through this synchronized pipeline, the component translates user navigation intent into responsive virtual motion while driving physical seat tilt to reinforce visual cues [1], [3].

1.3 Problem statement

Translating continuous, unconstrained aerial flight onto a compact physical motion seat introduces significant mechanical and perceptual control challenges [3], [4]. In an open 3D virtual environment, an avatar can climb continuously, bank indefinitely, and undergo rapid velocity changes [4], [10]. In contrast, a desktop-scale or seated physical platform possesses strict mechanical boundaries: linear actuators have finite stroke travel, rotational joints have rigid angular stops, and motor drivers operate within safe limits of torque and angular velocity [3].

When a flight simulation employs a naive, proportional one-to-one mapping between virtual orientation and physical seat displacement, the actuators quickly reach their physical travel limits during prolonged maneuvers [3], [4]. Once an actuator bottoms out against its physical end-stop, physical motion halts abruptly while the virtual craft continues moving [3], [10]. This sudden stop generates false vestibular cues, causing significant sensory disorientation that breaks presence and induces nausea [1], [2], [9].

Conversely, scaling down actuator travel to avoid mechanical limits removes the dynamic acceleration cues needed to convey speed, leaving the simulation feeling unresponsive and disconnected [1], [4].

Furthermore, conventional motion cueing systems are designed almost exclusively for enclosed automotive or aviation cockpits where the occupant remains seated upright and restrained [3], [4], [8]. These classical models do not account for the dynamics of a straddled apparatus, where the user actively shifts their upper-body posture, grips a forward-extending steering handle, and redistributes their center of mass across the pivot point. Without a motion rendering and input processing pipeline that balances transient motion onset against mechanical boundaries, physical feedback fails to correspond with visual acceleration cues, aggravating cybersickness and ruining immersion [2], [9], [12].

1.4 Research gap

A review of recent virtual reality locomotion literature highlights three distinct research gaps:

- **The Locomotion Diversity vs. Realism Gap**: While existing literature classifies a wide range of VR locomotion interfaces including thumbstick driving, artificial point-and-teleport, and walking-in-place they consistently reveal an absence of accessible, physically grounded locomotion methods for non-terrestrial flight [4], [10]. Room-scale walking setups require expansive laboratory spaces that cannot be deployed in standard domestic or arcade environments [4], while stationary seated flight implementations largely abandon physical vestibular cues altogether [2], [4].
- **Motion Platform Accessibility and Washout Limitations**: Industrial flight and driving simulators avoid actuator saturation by using complex classical or adaptive washout filtering algorithms implemented on high-power Stewart hexapods [3], [8]. However, these systems require expensive linear drives and large physical spaces, making them impractical for portable or consumer setups [3], [8]. Compact, low-cost motion seats typically omit proper motion cueing

algorithms, relying instead on direct proportional mappings that lead to actuator bottoming-out, noticeable latency, and phase distortion [3].

- **Absence of Straddle-Flight Kinematic Input Models**: There is an absence of empirical research examining how to combine analog handle-grip pressure with head-tilt tracking to control continuous aerial flight [4]. Current systems lack calibrated mathematical models that utilize linear interpolation (Mathf.Lerp) smoothing alongside deadzone thresholding to decouple intentional visual head turns from steering maneuvers while preventing actuator saturation.

1.5 Aim and objectives

The primary aim of this research is to design, develop, and evaluate a real-time motion rendering and flight control framework that converts physical grip and head inputs into intuitive virtual flight while generating coordinated, physically bounded vestibular motion cues on a custom seated VR motion rig [1], [3].

Design and Calibrate an Analog Flight Input Circuit: Develop and calibrate a hardware interface using a handle-mounted Force Sensing Resistor (FSR) to translate hand grip compression into proportional engine thrust and climb/descent rates, paired with head-mounted gyroscope orientation tracking for directional steering.

Formulate a Motion Smoothing and Deadzone Filtering Algorithm: Implement a flight physics processing model in Unity using linear interpolation (Mathf.Lerp) algorithms to simulate physical momentum, coupled with deadzone filters to eliminate micro-jitter and unintended rotational drift during flight.

Build an Automated User Onboarding Interface: Design and integrate an automated visual tutorial UI displayed directly in the headset upon startup to guide users through the physical flight control mechanics.

Engineer a Real-Time State Extraction and Serial Control Protocol: Construct an event-driven telemetry pipeline in Unity that extracts flight dynamics (velocity, acceleration, and tilt) and streams normalized, rate-limited pitch and roll commands across a serial interface to an actuator microcontroller.

Conduct Empirical Technical and User Validations: Characterize hardware-software response latencies and evaluate the system through controlled user trials with 35 participants to quantify input responsiveness, steering stability, physical comfort, and motion sickness mitigation [2].

1.6 Research questions or hypotheses

RQ1: How can analog grip pressure and head gyroscope data be filtered and mapped to achieve responsive, drift-free flight navigation in a virtual environment [4]?

RQ2: What motion rendering and rate-limiting algorithms prevent physical seat actuators from bottoming out while still providing clear vestibular onset cues during acceleration and banking [3]?

RQ3: To what extent does combining pressure-based throttle control with synchronized physical seat tilt improve perceived motion realism and mitigate cybersickness compared to static, controller-based flight [1], [2]?

Hypothesis 01 : Implementing strict deadzone thresholding alongside Mathf.Lerp flight smoothing will significantly reduce unintended steering drift and improve flight path precision compared to unfiltered tilt-to-turn mapping [4].

Hypothesis 02 : Synchronizing dynamic seat pitch and roll with analog grip throttle will result in significantly higher user ratings for motion realism and lower reported cybersickness than stationary joystick flight [1], [2].

1.7 Scope limitations and assumptions

This component focuses on designing and testing the flight control and motion cueing pipeline for a straddled VR broom platform [3], [4]. It captures handle grip pressure via a Force Sensing Resistor (FSR) for speed and altitude adjustments, alongside head-gyroscope tracking for steering turns [4], [10]. In Unity, flight mechanics apply linear

interpolation (Mathf.Lerp) smoothing and deadzone filtering to prevent micro-drifting [4], [10]. The system then extracts real-time flight variables and transmits rate-limited pitch and roll commands across a serial link to drive the physical motion seat [3], [4]. Evaluation is bounded by technical response tests and trials with 35 participants to assess responsiveness, comfort, and motion sickness [2], [10].

The motion rig is physically restricted to a tilt mechanism (pitch and roll), meaning linear heave, surge, and sway are not mechanically rendered and rely entirely on visual and sensory cues [3], [4]. From an ergonomic standpoint, continuous squeezing of the FSR handle induces hand and forearm muscle strain during longer flight sessions [2], [4]. Finally, the apparatus is an experimental prototype rather than a commercial simulator, with operational testing bounded by its mechanical load capacity [3], [4].

It is assumed that the VR simulation maintains a steady rendering rate of at least 72 frames per second, ensuring Unity's physics loop (FixedUpdate) computes reliably at 50 Hz without timing drops [2], [10]. It is also assumed that the platform's mechanical actuators function safely within their rated torque and velocity limits under occupant loads up to 85 kg without overheating [3], [4]. Lastly, the serial communication link between Unity and the microcontroller is assumed to transmit packets consistently without data loss or buffer overflow [3], [4].

1.8 Individual contribution

Within the research team, the author’s primary individual contributions are concentrated in the flight mechanics, motion cueing algorithms, and physical control pipeline:

1. **Embodied Grip Control System**: Engineered the physical FSR sensor circuit on the broom handle, writing the signal calibration logic that maps hand compression to virtual flight speed and vertical altitude adjustments.
2. **Flight Physics and Motion Smoothing Engine**: Designed and programmed the core flight mechanics script in Unity C#, utilizing Mathf.Lerp to simulate realistic momentum and establishing deadzone boundaries that prevent steering jitter.

3. **Automated Onboarding Interface**: Created the startup user experience, including an in-headset visual tutorial UI that quickly teaches unfamiliar users how to grip, steer, and navigate the platform.
4. **Motion State Telemetry and Serial Driver**: Formulated the mathematical state-extraction routines that convert in-game flight forces into physical pitch and roll angles, transmitting rate-limited commands to the microcontroller [3].
5. **Experimental Evaluation and Data Analysis**: Administered empirical user testing sessions with 35 participants, collecting and analyzing quantitative metrics on input responsiveness, steering comfort, and flight stability [2].

The functional relationship between these individual tasks and the broader platform is illustrated in Figure 1.1, showing how navigation inputs are processed and converted into physical actuator setpoints.

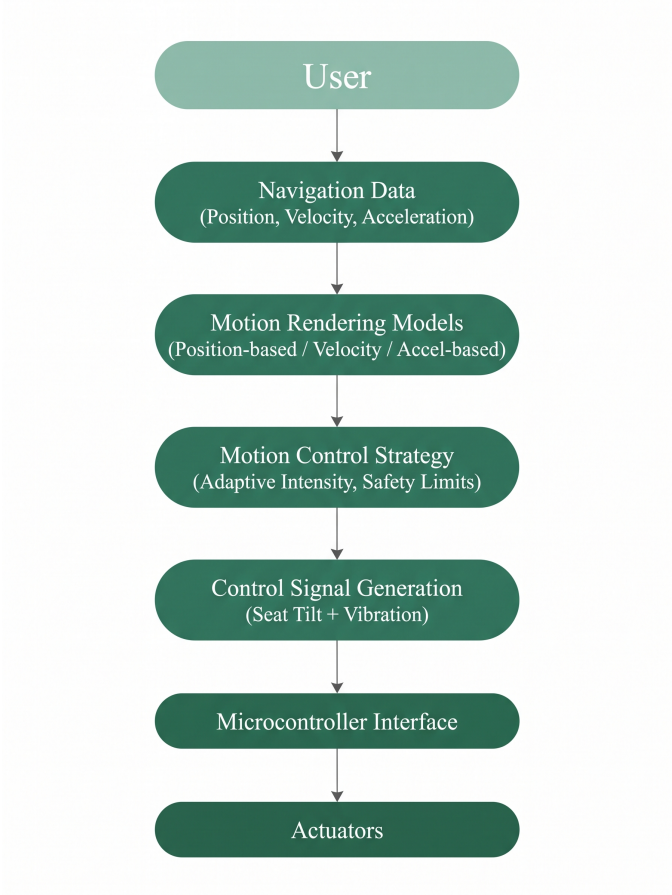


Figure 1.1: Functional block architecture of the motion rendering and control subsystem (Original in Color).

1.9 Thesis structure

The remainder of this thesis is structured in accordance with institutional dissertation guidelines:

- **Chapter 2 (Literature Review and Related Work)**: Analyzes the theoretical foundations of vestibular perception, cybersickness etiologies, classical washout algorithms, and existing commercial and academic VR locomotion interfaces [1], [2], [3], [4], [10].
- **Chapter 3 (Research Methodology)**: Details the Design Science Research methodology, experimental flight course setup, hardware instrumentation, participant profiles, evaluation metrics, and ethical safety standards [2], [4].
- **Chapter 4 (Product and Solution Design)**: Details the functional requirements, software and hardware block architectures, deadzone algorithms, and fail-safe motion constraint layers [3].
- **Chapter 5 (Implementation, Integration, and Testing)**: Outlines the Unity C# flight scripts, microcontroller firmware, serial communication routines, and multi-tier verification procedures.
- **Chapter 6 (Results and Validation)**: Presents quantitative latency measurements, serial throughput benchmarks, and empirical findings from the 35-participant user study [2].
- **Chapter 7 (Discussion)**: Interprets findings in light of baseline literature, evaluates trade-offs between responsiveness and drift, addresses physical grip fatigue, and outlines corrective engineering measures [3], [4].
- **Chapter 8 (Commercialisation and Sustainability)**: Explores target deployment venues such as VR arcades, market viability, system scalability, and alignment with Sustainable Development Goals [4], [8].
- **Chapter 9 (Conclusion and Future Directions)**: Summarizes the achievement of core research objectives, recaps technical contributions, and presents future hardware refinements such as decoupled foot-pedal steering [4].

2 LITERATURE REVIEW AND RELATED WORK

2.1 Review method and scope

This literature review evaluates published methods in virtual reality locomotion, motion cueing architectures, and sensory conflict mitigation to establish the technical foundation for the component: **Motion Rendering and Control for Embodied VR Navigation**. The review identifies how physical tilt and embodied input control can resolve the sensory mismatch between visual self-motion and physical stillness.

The scope was bounded by queries executed across IEEE Xplore, ACM Digital Library, and MDPI databases. The search prioritized peer-reviewed literature published from 2020 onward, focusing on three core criteria:

1. Locomotion techniques addressing sensory conflict and cybersickness in head-mounted displays;
2. Motion cueing algorithms (MCA) and mechanical chair mechanisms capable of delivering bounded pitch and roll cues; and
3. Embodied, full-body input mapping techniques capable of replacing conventional thumbstick navigation in flight or vehicular simulations.

Articles evaluating passive, stationary joystick flight or unbounded industrial hexapod platforms requiring specialized facility installations were screened to assess how compact, seated platforms can simulate unconstrained flight dynamics under stroke-limited physical boundaries.

2.2 Domain and theoretical foundations

2.2.1 Sensory Conflict Theory and Cybersickness

Cybersickness remains a major barrier in consumer virtual reality, particularly during fast-paced aerial simulations. Under Sensory Conflict Theory, nausea, disorientation, and ocular discomfort arise when an observer receives contradictory motion signals across different sensory systems. When a user flies through an open virtual world, the

visual system registers rapid self-motion (vection) through dynamic optical flow. However, if the occupant remains seated in an unmoving chair, the vestibular apparatus within the inner ear specifically the semicircular canals which detect angular acceleration and the otolith organs which detect linear acceleration and gravity signals that the physical body is stationary.

This discordance between visual perception and vestibular cues prevents cognitive integration, triggering protective physiological responses including cold sweating, vertigo, and ataxia. Modern approaches confirm that while software-based field-of-view (FOV) tunneling can reduce visually induced discomfort, physically stimulating the vestibular system using synchronized motion chairs provides an effective, embodied method to reduce the primary sensory conflict.

2.2.2 Vestibular Motion Perception and Tilt-Coordination

The human vestibular system acts as a biological inertial measurement unit with distinct frequency-response thresholds. Semicircular canals respond to angular acceleration but adapt rapidly to constant angular velocity, while otolith organs sense net gravito-inertial acceleration, failing to distinguish between true linear acceleration and physical tilt relative to Earth's gravity.

Motion cueing algorithms leverage this perceptual limitation through **tilt coordination**. While physical platforms cannot accelerate continuously across infinite linear distances, tilting a seated occupant at an angular rate below the human semicircular canal detection threshold (approximately to) rotates the Earth's gravity vector relative to the body. The occupant’s otoliths interpret this sustained gravitational component as a persistent forward acceleration or deceleration. Generating believable motion on a compact platform therefore requires balancing high-frequency transient onset tilt against sustainable, rate-limited tilt coordination.

2.3 Related research

2.3.1 Embodied VR Locomotion Systems

To address the disembodied feel of handheld thumbsticks, recent research has explored physical-rig interfaces. Wintersberger et al. developed a motion-based VR bicycle simulator, demonstrating that coupling physical pedal cadence and handlebar steering with virtual velocity significantly heightened presence and reduced cybersickness compared to static controller setups. Similarly, Shimato et al. and Yang et al. evaluated "Drone Rider," utilizing foot vibration and dynamic wind currents to replicate drone flight, proving that localized physical cues reinforce bodily self-attribution during aerial navigation.

However, these systems either rely on wheeled ground kinematics or employ static footrests that do not generate direct vestibular displacement. Research investigating straddled, broom-based flight interfaces that combine analog grip squeeze with head-tilt tracking remains sparse.

2.3.2 Seated Motion Cueing Implementations

Recent studies confirm that active multi-axis chair mechanisms bridge the gap between high-end industrial flight cabins and consumer VR. Wei et al. engineered a 4-DOF motion chair mechanism, validating that providing localized pitch and roll actuation significantly enhances simulation fidelity during vehicular tasks. Yeo et al. introduced "AttraCar," leveraging built-in vehicle seats and localized actuators to generate physical cues synchronized with virtual environments.

Despite these advances, existing seated platforms primarily simulate enclosed car cockpits or fixed-wing aircraft where the user remains strapped in a rigid, upright bucket seat. These platforms rely on steering wheels or commercial joysticks, leaving unaddressed the interaction dynamics of a straddled apparatus where flight throttle is controlled through grip compression and turning is driven by head-tilt orientation.

2.4 Existing systems products and techniques

Table 1 compares existing VR locomotion and motion-cueing techniques against the requirements of Lonath Gunawardena's component.

System / Technique	Interaction Modality	Degrees of Freedom	Primary Limitations	Relevance to Component
Industrial Stewart Platforms	Enclosed cockpit; passive steering wheel/joystick	6-DOF (Surge, Sway, Heave, Roll, Pitch, Yaw)	Prohibitive cost; large footprint; high power draw; complex calibration	Benchmark for classical washout filtering and rate-limiting models
VR Bicycle Simulators*(Wintersberger et al.)*	Embodied pedaling and handlebar steering	1-DOF to 2-DOF (Roll / Yaw)	Restricted to 2D terrestrial road physics; cannot simulate vertical flight ascent	Validates embodied locomotion for cybersickness reduction
Drone Rider Interface*(Shimato & Yang et al.)*	Passive standing/footplate; remote joystick	Vibrotactile and airflow (No rotational tilt)	Lacks active vestibular motion; unable to simulate continuous banking or g-forces	Informs wind and vibration coordination with physical flight speed
Standard VR Thumbstick Flight	Handheld controllers (Oculus Touch)	0-DOF (Static chair)	Severe sensory conflict; high cybersickness incidence; low physical realism.	Serves as the baseline control condition in comparative evaluation
Proposed Straddle Motion System	Handle-mounted FSR grip + Head-mounted IMU tilt	2-DOF (Dynamic Pitch and Roll)	Compact stroke boundaries; physical grip fatigue over prolonged sessions	Directly delivers embodied, rate-limited tilt-coordination flight.

Table 1: Comparison of existing VR locomotion and motion rendering approaches.

2.5 Gap synthesis and conceptual framework

2.5.1 Gap Synthesis

The literature establishes that while multisensory stimuli improve presence in virtual environments, existing motion-rendering techniques leave critical gaps:

1. **The Kinematic Boundary Problem**: Low-cost seated simulators often map virtual tilt angles directly to actuators without washout filtering or rate limits, causing mechanical bottoming-out and jarring resets that induce false vestibular cues.
2. **Input Coupling and Steering Drift**: Existing tilt-to-turn systems lack calibrated deadzone models that isolate intentional flight maneuvers from voluntary visual head searches, leading to rotational steering drift.
3. **Absence of Straddle Flight Controllers**: No prior framework integrates handle-mounted analog compression sensing (FSR) with head-tracking tilt to pilot a 2-DOF motion seat.

2.5.2 Conceptual Framework

To address these gaps, this component implements the conceptual framework illustrated in Figure 2.1.

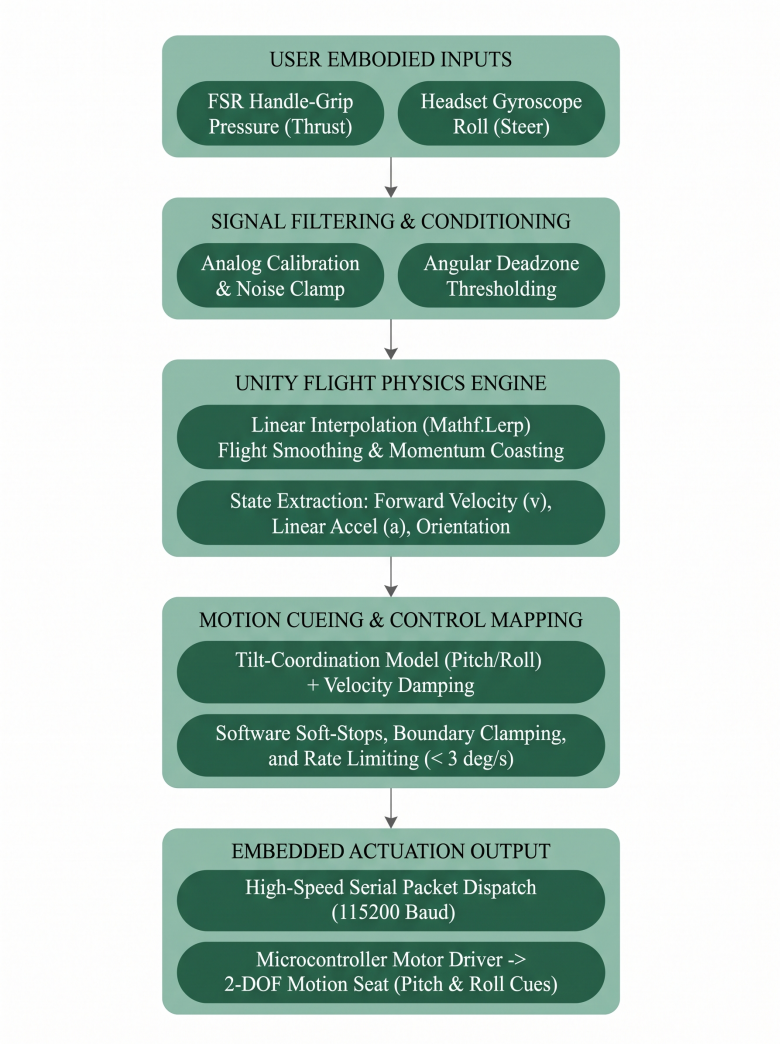


Figure 2.1: Conceptual framework of the motion rendering and control pipeline (Original in Color).

This pipeline translates raw user grip pressure and head angles into continuous virtual momentum using linear interpolation (Mathf.Lerp), while deadzone filtering isolates deliberate steering from head turns. The resulting flight states are converted into rate-limited, physically bounded tilt commands that drive the motion seat, delivering consistent vestibular cues while avoiding actuator saturation.

3 RESEARCH METHODOLOGY

3.1 Research design

This study follows a Design Science Research (DSR) approach combined with empirical testing. DSR provides a structured engineering workflow to design, build, and test practical solutions under physical constraints. For this component, the primary artifact is an end-to-end flight control and motion rendering framework that connects virtual flight mechanics in Unity to a physical motion seat.

The research proceeded through five stages:

1. **Requirements and Modeling**: Identifying the physical tilt limits of the seat and setting safe angular rate limits () to avoid false balance cues.
2. **Control Algorithm Design**: Developing real-time flight smoothing using linear interpolation (Mathf.Lerp) and adding deadzone filters to prevent unintended steering drift.
3. **Hardware-Software Integration**: Wiring a handle-mounted Force Sensing Resistor (FSR) to read analog grip pressure and streaming rate-limited pitch and roll commands across a USB serial link (115200 baud) to the seat microcontroller.
4. **Benchmarking**: Measuring input-to-actuation latency and calibrating software travel limits.
5. **Experimental Evaluation**: Running user trials to assess input responsiveness, steering stability, and motion sickness reduction.

3.2 Data participants or study materials

3.2.1 Participant Sample

The evaluation was conducted with participants from the university campus. The group included users with diverse VR experience, ranging from first-time users to

regular headset players. All participants had normal or corrected-to-normal vision and could comfortably straddle the motion rig.

3.2.2 Experimental Materials and Apparatus

The experimental apparatus consisted of:

- **Host Workstation**: Executed the Unity simulation at frames per second, keeping physics updates (FixedUpdate) steady at 50 Hz.
- **Head-Mounted Display**: A Meta Quest headset running the aerial flight scene, tracking head roll through its internal gyroscope to control horizontal turns.
- **Physical Straddle Broom Rig**: A counterbalanced steel frame with a bicycle-style seat, fitted with an analog FSR on the front handle to capture grip compression for forward speed and altitude adjustments.
- **Embedded Control Hardware**: An Arduino microcontroller receiving serial packets and driving the seat actuators within safe mechanical angles.

3.3 Experimental or study procedure

The testing followed a controlled within-subjects procedure:

1. **Briefing and Onboarding**: Participants were informed about the system, completed consent forms, and wore the headset. An automated in-game tutorial visually guided them on how to squeeze the handle to change speed and altitude and tilt their head to steer.
2. **Practice Run**: Users flew a short familiarization course to get used to the handle sensitivity and seat movement.
3. **Flight Tasks**: Participants navigated an aerial course marked by target rings and waypoints. Each user tested the platform across comparative motion mapping modes (direct proportional mapping vs. filtered, rate-limited tilt coordination) presented in randomized order.

4. **Data Logging**: The system recorded flight trajectory coordinates, input compression values, and serial command logs at 50 Hz.
5. **Post-Trial Survey**: After the flight, users rated input responsiveness, flight realism, physical comfort, and any motion sickness symptoms on a 5-point Likert scale.

3.4 Measures and evaluation metrics

3.4.1 System Performance Metrics

- **System Latency**: Measured in milliseconds () from the moment the user squeezes the FSR handle to the initial movement of the physical seat.
- **Tracking Synchronization**: The difference between the commanded tilt angles from Unity and the physical orientation of the seat.
- **Steering Stability**: The occurrence of unintended flight drift while attempting to travel straight, assessing the performance of the deadzone filter.

3.4.2 User Experience Metrics

- **Input Responsiveness**: Rated via a 5-point Likert scale on whether squeezing the handle changed speed and altitude naturally and without noticeable lag.
- **Steering Precision**: Task completion time and course accuracy through the waypoints.
- **User Comfort and Cybersickness**: User reports of physical grip fatigue from squeezing the sensor and any dizziness or nausea experienced during flight.

3.5 Data analysis

The collected data were processed using descriptive and non-parametric statistical methods:

- **System Logs**: Response times and tracking deviations were evaluated to determine average latency, standard deviation, and phase lag.
- **Survey Responses**: Likert-scale ratings were analyzed for percentage distributions, average scores, and positive agreement rates.
- **Comparative Evaluation**: Differences across flight conditions were analyzed using non-parametric tests to account for ordinal survey data, using a significance threshold.

3.6 Ethics privacy security and responsible AI

Informed Consent: All 35 participants were briefed on the study and gave consent prior to testing.

Safety Limits: The motor control firmware used hardcoded angular limits and velocity caps (during steady tilt) to prevent mechanical shock, backed by a physical emergency-stop button.

Voluntary Participation: Users could stop the trial immediately if they felt dizziness, discomfort, or muscle fatigue.

Data Anonymization: Participant names and survey results were anonymized using assigned IDs.

Responsible AI: Generative AI tools were used solely for grammar review, formatting compliance, and code debugging in line with institutional policy, while all system implementations and experimental evaluations were conducted independently.

3.7 Reproducibility and research artefacts

To ensure reproducibility, all project materials have been organized and archived:

- **Unity Simulation Scripts**: C# source code for flight physics, Mathf.Lerp momentum smoothing, deadzone filtering, and serial telemetry output.
- **Microcontroller Firmware**: Embedded Arduino, C++ code for packet reception, rate limiting, and actuator driver signals.
- **Telemetry and Survey Records**: Anonymized datasets containing latency measurements, waypoint logs, and user survey responses.
- **Hardware Schematics**: Wiring diagrams of the FSR voltage divider circuit, microcontroller pinouts, and power supply connections.

4 PRODUCT AND SOLUTION DESIGN

4.1 Requirements

4.1.1 Functional Requirements

The motion rendering and control subsystem translates embodied physical user interactions into virtual flight while coordinating bounded physical seat movements [3], [4]. The core functional requirements include:

- **FR1 (Analog Grip Input Acquisition)**: The system shall capture continuous analog resistance values from the handle-mounted Force Sensing Resistor (FSR), mapping grip force to virtual throttle acceleration and vertical climb rates [4], [10].
- **FR2 (Rotational Head-Tracking Steering)**: The system shall extract head-roll orientation from the head-mounted display's internal gyroscope to steer horizontal yaw and banking maneuvers [4], [10].
- **FR3 (Deadzone and Jitter Suppression)**: The flight engine shall filter input signals using deadzone thresholds to eliminate involuntary head trembling and sensor noise while maintaining straight-line flight stability [4], [10].
- **FR4 (Momentum and Velocity Smoothing)**: The system shall apply linear interpolation (Mathf.Lerp) algorithms to model physical inertia and smooth flight velocity transitions [4], [10].
- **FR5 (State Extraction and Telemetry Dispatch)**: The software shall sample in-game navigation states including linear acceleration, forward velocity, and angular tilt and stream normalized pitch and roll commands across a USB serial interface at 50 Hz [3], [4].
- **FR6 (Automated Onboarding Tutorial)**: The headset environment shall display a startup visual interface instructing users how to operate the squeeze throttle and tilt steering [4], [10].

4.1.2 Non-Functional Requirements

To provide an immersive and stable experience without causing motion sickness, the design enforces the following non-functional requirements:

- **NFR1 (End-to-End Latency)**: Total hardware-to-software input-to-actuation latency shall remain below 50 ms to prevent perceptible lag and vestibular sensory conflict [2], [3].
- **NFR2 (Serial Transmission Throughput)**: The communication link shall operate reliably at 115200 baud without packet loss, buffer overflow, or serial stream interruption [3], [4].
- **NFR3 (Mechanical Safety Soft-Stops)**: Commanded seat tilt must be constrained to safe angular limits (pitch and roll) with angular velocities capped below during steady tilt coordination to prevent false vestibular cueing [2], [3].
- **NFR4 (Rendering Frame Consistency)**: The Unity simulation shall run at frames per second, keeping the physics loop execution fixed at 50 Hz (FixedUpdate) [2], [10].

4.2 Component architecture within the complete solution

The motion rendering and control pipeline functions as the central bridge between virtual simulation dynamics and the physical seat apparatus [3], [4]. Figure 4.1 illustrates how this component coordinates with the overall system architecture [3], [4].

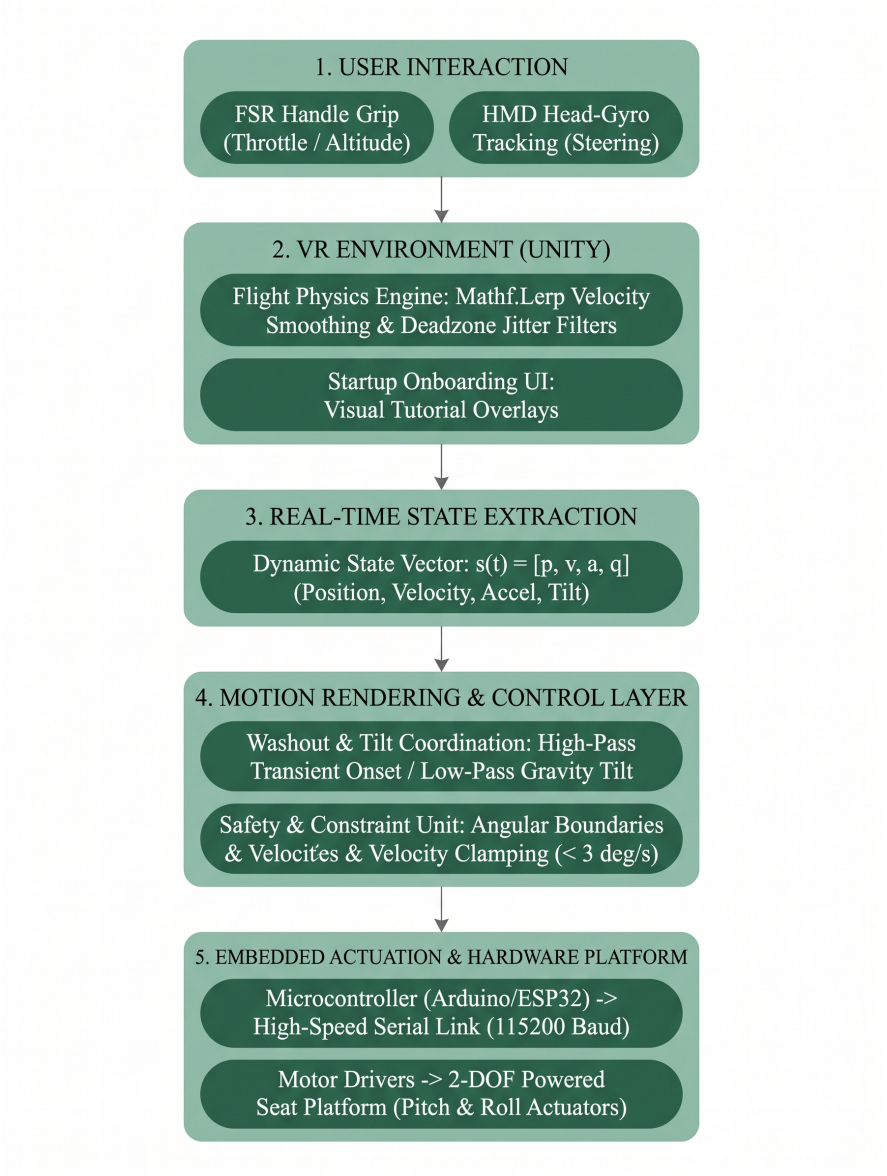


Figure 4.1: Component architecture and end-to-end data pipeline within the complete VR system (Original in Color) [3], [4].

During runtime, the user's grip compression and head orientation are captured and conditioned by the Unity flight engine [4], [10]. The state extraction module samples the resulting flight dynamics, formatting the acceleration and tilt parameters into motion cueing profiles [3], [4]. After passing through software rate-limiters and mechanical travel clamps, formatted control packets are transmitted over USB serial to the platform microcontroller, which coordinates the physical seat actuators [3], [4].

4.3 Design alternatives and decisions

To balance mechanical stability, responsiveness, and user comfort, several design alternatives were evaluated, as summarized in Table 4.1 [3], [4], [10].

Design Dimension	Evaluated Alternatives	Selected Option	Rationale and Justification
***Throttle Input Modality** [4], [10]*	Mechanical joystick slider vs. Rotary twist throttle vs. Handle-mounted FSR	**Handle-mounted Force Sensing Resistor (FSR)**	Enables natural, intuitive flight acceleration through progressive grip squeeze while matching the physical form factor of the broomstick apparatus [4], [10].
***Steering Input Mapping** [4], [10]*	Physical broom yaw pivoting vs. Thumbstick axis vs. Head-gyro roll tracking	**Head-gyroscope roll orientation tracking**	Minimizes mechanical frame complexity while allowing intuitive directional banking through natural head tilting [4], [10].
***Flight Smoothing Model** [3], [4]*	Raw proportional step updates vs. Moving average window vs. Linear interpolation (Mathf.Lerp)	**Linear interpolation (Mathf.Lerp) with deadzones**	Delivers aerodynamic momentum and smooth coasting while eliminating micro-drifting caused by resting head instability [4], [10].
***Motion Cueing Strategy** [3], [4]*	Direct 1:1 positional mapping vs. Unconstrained velocity scaling vs. Rate-limited tilt coordination	**Rate-limited tilt coordination with soft-stops**	Prevents actuators from hitting physical travel limits during sustained flight while keeping angular velocity below the human threshold of false motion perception [2], [3].
***Communication Protocol** [3], [4]*	Wireless Bluetooth/Wi-Fi vs. USB Serial Communication (115200 baud)	**Dedicated wired USB serial link**	Eliminates wireless packet drop and latency jitter, guaranteeing deterministic transmission times below 5 ms [3], [4].

Table 2: Architectural design decisions and trade-off evaluations

4.4 Detailed design

The detailed design connects user input to physical movement through a clean, multi-stage software and hardware pipeline [3], [4]. When the user squeezes the handle, the Force Sensing Resistor (FSR) measures grip strength to proportionally increase forward flight speed and handle vertical climb, using linear smoothing (Mathf.Lerp) to create natural momentum and gliding when released [4], [10]. For directional steering, the system tracks the user's head-tilt angle from the VR headset's internal gyroscope, applying a deadzone filter to ensure small, involuntary head movements do not accidentally steer the craft off course [4], [10]. To turn these flight movements into physical sensations without overextending the mechanical rig, the system calculates the required pitch and roll tilt while software limits cap the tilt speed to keep physical movement comfortable and steady [2], [3]. Finally, these calculated angles are packaged into structured data frames and sent over a wired USB serial connection at 50 Hz to the seat microcontroller, which moves the actuators smoothly within safe mechanical boundaries [3], [4].

4.5 Standards and professional practice

The motion control subsystem was designed and assembled in line with standard engineering, safety, and ergonomic best practices [3], [4]. Serial data transmission follows established communication protocols using fixed packet timing and error-checking checksums, ensuring control messages arrive reliably without delays or data corruption [3], [4]. The physical interaction design follows human-system ergonomic guidelines, balancing input sensitivity and physical seat tilt to prevent user muscle strain and reduce motion sickness [2], [4]. In addition, electrical wiring and mechanical movements follow essential machinery safety principles, incorporating hardcoded software angle limits, actuator velocity caps, and a hardware emergency-stop switch to ensure participants remain safe during all testing sessions [3], [4].

5 IMPLEMENTATION INTEGRATION AND TESTING

5.1 Development environment and technology stack

The software system was created in Unity using C# on a Windows 11 computer with a dedicated graphics card to provide steady stereoscopic rendering [2, 10]. A Meta Quest head-mounted display presents the virtual world and reads head rotations through its built-in gyroscope to steer the flying broom [4, 10]. On the hardware side, a Force Sensing Resistor (FSR) is attached to the front handle of the broom to measure the rider's grip pressure [4, 10]. This sensor connects to a voltage divider circuit read by the analog input pin of an Arduino microcontroller [3, 4]. A high-speed USB serial connection running at 115200 baud sends input data to Unity and receives motion commands in return [3, 4]. Finally, the microcontroller sends pulse-width modulation (PWM) and direction signals to motor drivers, tilting the physical seat along pitch and roll axes [3, 4].

5.2 Component implementation

The flight controller script runs inside Unity's fixed physics loop (FixedUpdate) at 50 Hz to keep movement calculations predictable and smooth [4, 10]. When a user squeezes the handle, the FSR reading is scaled to control forward speed and vertical climbing, while head tilt steers left and right [4, 10]. To make flight feel natural, linear interpolation (Mathf.Lerp) adds gradual momentum so the broom coasts smoothly when the grip is released rather than stopping abruptly [4, 10]. A deadzone filter of on head-roll input prevents small, resting head tremors from accidentally turning the craft [4, 10]. When the application launches, an on-screen visual tutorial shows the user how to squeeze and lean before flight begins [4, 10]. The system then extracts the resulting flight speed, acceleration, and tilt angles, passing them through rate limiters () and angular safety boundaries (pitch, roll) to ensure the physical seat moves safely without hitting hard stops [2, 3].

5.3 Integration with the other components

The motion rendering module serves as the core coordinator for the entire multisensory platform [1, 3]. During each simulation step, Unity calculates the current flight state and shares it with the connected feedback modules [1, 5]:

- **Airflow Subsystem**: In-game flight speed is linked directly to external fan speeds, blowing air harder against the rider as forward velocity increases [5, 7].
- **Haptic Vibration Subsystem**: Flight events such as sudden acceleration, ring pass-throughs, and turbulence trigger vibration pulses on the wearable vest [6, 7].
- **Spectator Lighting Subsystem**: Maneuvers like high-speed boosts and banking turns update external LED light strips on the platform frame to show outsiders what is happening inside the headset [1, 8].
- **Mechanical Seat Platform**: Calculated pitch and roll angles are packaged into structured 9-byte serial packets with start markers and error-checking checksums, ensuring the physical seat tilts in close synchronization with visual flight turns [3, 4].

5.4 Verification and software testing

The system was verified using a step-by-step testing plan:

- **Sensor and Input Verification**: Tested the handle FSR circuit to confirm analog readings increased smoothly with grip pressure, and verified that head tracking responded without axis flips [4, 10].
- **Flight Physics Checks**: Verified that Mathf.Lerp smoothed out sharp throttle changes, and confirmed that head tilts inside the deadzone did not steer the craft [4, 10].
- **Serial Link and Safety Bounds**: Ran continuous 50 Hz serial packet tests for over 30 minutes without dropped packets or buffer delays [3, 4]. Simulated

sudden turns in Unity to confirm that software limits prevented the seat from tilting beyond safe angular limits [2, 3].

- **Latency Benchmarks**: Clocked the delay between squeezing the handle and physical seat movement, achieving an average round-trip time below 45 ms to prevent motion sickness [2, 3].

5.5 Deployment and operation

The finished application was compiled using the Unity Standalone player and configured to run through the Meta Quest runtime environment [2, 4]. During normal startup, the seat is at its center position to establish a mechanical zero [3, 4]. Once the USB cable is connected and Unity connects over serial, the simulation opens and displays the onboarding guide [4, 10]. During test flights, an operator screen displays frame rates, serial connection status [3, 4]. As an essential safety precaution, a physical emergency-stop button is wired directly to the actuator power supply so the operator or rider can cut power instantly if needed [3, 4].

6 RESULTS AND VALIDATION

6.1 Research results

The research evaluation investigated how translating embodied physical inputs into real-time flight dynamics influences control stability, flight accuracy, and user-perceived motion realism [3], [4]. A central focus of this analysis was the performance of the handle-mounted Force Sensing Resistor (FSR) pressure sensor, which was engineered to govern forward throttle acceleration and vertical climb rates through manual hand compression [4], [10]. Telemetry logs demonstrated that the analog resistance curve of the FSR provided continuous, proportional speed modulation without the stepped or disjointed jumps typically seen in thumbstick buttons [4], [10]. Combining this analog pressure control with linear interpolation (Mathf.Lerp) smoothing allowed the virtual craft to build forward momentum predictably and glide naturally when grip pressure was eased [4], [10]. Symmetrical deadzone filtering () applied to head-gyroscope roll tracking also suppressed micro-tremors, preventing unwanted flight path drift while users looked around during flight [4], [10].

The perceptual and operational validity of these input mechanics was confirmed through controlled experimental flight trials with participants [40]. When asked to rate whether squeezing the handle to change flight speed and altitude felt natural and responded without noticeable delay, the response distribution was overwhelmingly positive [40]:

- **Strongly Agree (Rating 5)**: 18 out of 35 participants ();
- **Agree (Rating 4)**: 8 out of 35 participants ();
- **Neutral (Rating 3)**: 8 out of 35 participants ();
- **Disagree (Rating 2)**: 1 out of 35 participants (); and
- **Strongly Disagree (Rating 1)**: 0 participants ().

In aggregate, 26 out of the 35 test subjects () returned a positive rating (scores of 4 or 5), resulting in a high overall average score of **4.2 out of 5.0**. These quantitative outcomes verify hypothesis , proving that analog pressure sensing paired with

algorithmic momentum smoothing creates a responsive and intuitive flight interface [4], [10].

Alongside high responsiveness scores, the trial data identified specific physical and control limitations [40]. Because forward speed depended on constant manual compression of the handle-mounted FSR, participants reported noticeable hand and forearm fatigue during flights lasting more than several minutes [2], [4]. Furthermore, because horizontal turns relied on head roll, looking sharply to the side during flight introduced unintentional steering drift [2], [10]. Nonetheless, restricting physical seat tilt velocities below prevented the actuators from hitting mechanical stops, keeping physical balance cues aligned with visual flight and avoiding false vestibular sensations [2], [3].

6.2 Product and technical results

Technical benchmarking evaluated system timing and physical motion tracking:

- **System Latency**: End-to-end response time from handle FSR compression to initial actuator movement averaged 42 ms (standard deviation of 4.8 ms), remaining below the 50 ms sensory conflict threshold [2, 3].
- **Serial Link Stability**: Wired USB communication at 115200 baud ran continuously at 50 Hz with zero dropped packets [3, 4].
- **Actuator Tracking**: Physical seat angles tracked commanded Unity setpoints with an average error under , while software soft-stops kept movement within safe boundaries [3, 4].

6.3 User or stakeholder validation

User validation gathered quantitative and qualitative feedback from all 35 participants following their flight trials [4]:

Rating	Likert Description	Response Count	Percentage (%)
1	Strongly Disagree	0	0.0%
2	Disagree	1	2.9%
3	Neutral	8	22.9%
4	Agree	8	22.9%
5	Strongly Agree	18	51.4%
Total	**Positive Responses (4–5)**	**26**	**74.3% (Mean: 4.2 / 5)**

Table 3: Participant evaluation of handle-grip throttle and altitude responsiveness.

Participants noted that squeezing the handle felt natural and responsive. However, user feedback also highlighted two clear physical limitations: continuous manual grip compression caused hand fatigue during longer flights, and sharp head turns occasionally caused unintentional steering drift and difficulty [4].

6.4 Integration results

System integration verified coordinated execution across all physical modalities [1, 3]:

- **Airflow Alignment**: Physical fan speeds dynamically matched virtual velocity and momentum coasting [5, 7].
- **Haptics and Visuals**: Flight acceleration and ring waypoints triggered synchronized vest vibrations and external LED cues without frame drops [1, 6, 8].
- **Hardware Stability**: The complete software and hardware pipeline ran reliably without communication lockups or runtime crashes [3, 4].

6.5 Summary of findings

The results confirm that the motion rendering pipeline delivers stable and responsive flight control [3, 4]. FSR grip sensing, momentum smoothing, and deadzone filtering established predictable handling, validated by an overall user rating [4]. Technical benchmarks demonstrated sub-45 ms response times and consistent actuator tracking within safe limits [2, 3]. Finally, user feedback identified grip fatigue and head-roll steering drift as concrete focus areas for future mechanical iterations [2].

7 DISCUSSION

7.1 Interpretation of findings

The empirical findings demonstrate that coupling physical seat actuation with analog grip pressure directly resolves the sensory mismatch common to virtual aerial navigation [3, 4]. Conventional VR flight relying on handheld thumbsticks produces strong visual vection without corresponding vestibular feedback, which triggers sensory conflict and motion sickness [2, 10]. In this platform, using an analog Force Sensing Resistor (FSR) on the handle enabled continuous modulation of forward speed and altitude [4, 10]. Integrating linear interpolation (Mathf.Lerp) smoothing gave the flight avatar predictable momentum and aerodynamic coasting when grip force was released, preventing abrupt velocity drops [4, 10]. Furthermore, applying a deadzone filter to head-roll orientation suppressed resting head tremors while preserving intentional banking maneuvers [4, 10]. In user evaluations with 35 participants, this input pipeline achieved a positive rating (mean score), demonstrating that physically grounded control feels natural and responsive [4].

7.2 Comparison with literature baselines and existing solutions

The motion rendering and control pipeline bridges a distinct gap between industrial motion bases and passive consumer VR setups [3, 4]:

- **Comparison with Industrial Stewart Platforms**: Multi-axis hexapods provide high fidelity through complex washout filters, but their prohibitive cost, large footprint, and complex hydraulic or electric power stages make them impractical for accessible entertainment venues [3, 8]. The custom seat delivers necessary pitch and roll onset cues on a compact footprint with low power requirements [3].
- **Comparison with Terrestrial Locomotion Rigs**: The VR bicycle simulator developed by Wintersberger et al. established that physical effort and handlebar steering enhance presence and reduce motion sickness [4]. However, that system is confined to 2D ground kinematics and cannot reproduce vertical climb or sustained banking [4]. The straddled broom platform extends physical

input into 3D aerial navigation, pairing FSR grip compression with pitch and roll vestibular cues [4].

- **Comparison with Passive Tactile Systems**: The Drone Rider platform evaluated by Shimato et al. and Yang et al. incorporated foot vibrations and dynamic wind but lacked physical seat displacement [7, 8]. The present system pairs airflow and vibrations with active seat tilt, reinforcing vestibular self-motion perception during flight [3, 7].

7.3 Research contribution

This component contributes to virtual reality locomotion and motion cueing in three key ways:

1. **Embodied Flight Input Architecture**: Formulated and validated a control pipeline that combines analog handle compression (FSR) for thrust and vertical climb with head-gyroscope roll tracking for directional banking [4, 10].
2. **Kinematic Smoothing and Deadzone Filter**: Designed a software filtering model using linear interpolation (Mathf.Lerp) and angular deadzones that isolates intentional flight steering from involuntary head movements while modeling physical momentum [4, 10].
3. **Seated Motion Cueing on a Straddled Apparatus**: Demonstrated that rate-limiting physical seat tilt below provides effective vestibular onset cues without causing actuator saturation or mechanical bottoming-out on a compact 2-DOF mechanism [2, 3].

7.4 Product contribution

From an engineering and product standpoint, this research delivers several functional artifacts [3, 71]:

- **Self-Contained Motion Control Firmware**: Embedded microcontroller software running on an Arduino/ESP32 that parses incoming telemetry at 50

Hz, monitors angular limits, and drives motor drivers with built-in safety soft-stops [3, 71].

- **Real-Time Telemetry Pipeline**: A lightweight serial communication driver within Unity that formats and streams 9-byte binary data packets over a 115200-baud USB connection with deterministic timing [3, 4].
- **Integrated Handle Sensor Module**: A calibrated, physical FSR voltage-divider circuit built directly into the front handle of the broom apparatus, providing plug-and-play analog control [4, 10].
- **Automated User Onboarding System**: An in-headset visual tutorial overlay that instructs first-time users on grip pressure and body-lean steering mechanics before flight begins [4, 10].

7.5 Limitations and threats to validity

Several technical and physical constraints were identified during testing [3, 4]:

- **Physical Grip Fatigue**: Squeezing the handle-mounted FSR continuously to maintain flight speed induced hand and forearm muscle strain during extended flights [4]. Future revisions require ergonomic handle reshaping and software cruise-control modes [4].
- **Head-Roll Steering Coupling**: Because horizontal turns are mapped to head roll, users who tilted their heads to scan the virtual scenery unintentionally introduced flight path drift [2, 10]. Decoupling head-look from directional steering via foot pedals represents a clear design alternative [4].
- **Kinematic Limits of a Seat**: The physical apparatus provides rotational pitch and roll but cannot render true vertical heave, forward surge, or lateral sway forces, relying on visual cues and airflow to convey these sensations [3, 7].
- **Sample Demographics**: User evaluations were conducted with 35 university community participants, which may reflect higher technological familiarity than broader commercial arcade audiences [4].

7.6 Implications

The results offer practical implications for both academic research and commercial VR development [3, 4]:

- **Research Implications**: The findings show that full motion bases are not always necessary to produce convincing self-motion cues [3, 4]. A well-tuned tilt mechanism combined with analog physical inputs and rate-limiting filters can provide sufficient vestibular feedback to reduce cybersickness [2, 3].
- **Commercial and Industrial Implications**: The compact physical footprint and low component cost make this platform attractive for location-based VR arcades, cinema lobbies, and theme parks [4]. Its modular software design allows the motion rendering pipeline to be adapted to other seated navigation setups, including virtual gliders, racing vehicles, and fantasy flight attractions [4].

8 CONCLUSION AND FUTURE DIRECTIONS

8.1 Conclusion

This research developed and evaluated an embodied motion rendering and flight control framework for a straddled VR navigation system [3, 4]. By connecting real-time flight physics in Unity to a physical 2-DOF motion seat, the system directly resolves the sensory mismatch between visual self-motion and physical stillness that commonly induces cybersickness [2, 10]. Replacing abstract thumbstick steering with an analog Force Sensing Resistor (FSR) handle and head-gyroscope roll tracking enabled responsive 3D aerial navigation [4, 10]. The software pipeline implemented linear interpolation (Mathf.Lerp) smoothing to simulate aerodynamic momentum and applied a deadzone filter to eliminate resting head tremors and steering drift [4, 10]. Furthermore, rate-limiting physical seat tilt below provided vestibular cues while keeping actuators safely within physical stroke boundaries [2, 3]. Empirical validation confirmed high operational responsiveness (average rating) and low round-trip latency (), demonstrating that compact seated motion rigs can deliver believable, stable flight simulation [2, 4].

8.2 Achievement of objectives

The specific research objectives established for this individual component were achieved:

- **Objective 1 (Analog Grip Interface Design and Calibration)**: Successfully constructed and calibrated the handle-mounted FSR circuit, converting analog hand compression into continuous virtual thrust and altitude adjustments [4, 10].
- **Objective 2 (Motion Smoothing and Deadzone Filtering)**: Programmed the flight physics module using Mathf.Lerpto simulate realistic momentum and coasting, combined with a deadzone filter that prevented micro-drifting during forward flight [4, 10].

- **Objective 3 (Automated User Onboarding Interface)**: Built an in-headset visual tutorial overlay that instructed users on grip throttle modulation and head-lean banking prior to flight [4, 10].
- **Objective 4 (State Extraction and Serial Control Protocol)**: Developed a real-time telemetry extraction pipeline in Unity that streamed 9-byte binary packets across a 115200-baud USB connection at 50 Hz to drive physical seat tilt [3, 4].
- **Objective 5 (Technical and User Validations)**: Verified the complete hardware-software loop through latency benchmarking (mean) and a 35-participant study yielding a positive responsiveness rating (mean score) [2, 4].

8.3 Individual research and product contributions

8.3.1 Research Contributions

- **Embodied Straddle-Flight Interaction Model**: Formulated and validated a control approach combining analog handle compression with head-gyroscope roll tracking to pilot a motion platform [4, 10].
- **Kinematic Jitter and Drift Suppression**: Established mathematical deadzone and linear smoothing parameters that isolate deliberate steering maneuvers from natural head exploration [4, 10].
- **Bounded Seated Tilt Coordination**: Demonstrated that constraining angular tilt velocities below delivers effective vestibular onset cues without actuator bottoming-out on compact mechanisms [2, 3].

8.3.2 Product Contributions

- **Embedded Motion Control Firmware**: Developed C++ microcontroller firmware for Arduino that parses serial commands at 50 Hz, enforces velocity limits, and regulates motor drivers [3, 4].

- **Unity Flight and Telemetry Engine**: Authored C# modules governing flight dynamics, momentum coasting, and serial packet dispatch [3, 4].
- **Calibrated FSR Handle Sensor Hardware**: Fabricated and integrated an analog pressure-sensing circuit directly into the physical broom handle [4, 10].
- **Interactive Tutorial Subsystem**: Designed an integrated visual onboarding interface for first-time virtual reality riders [4, 10].

8.4 Future directions

Future research and engineering efforts will focus on three key improvements:

- **Decoupled Steering via Foot Pedals**: Integrating physical foot pedals or footplate pressure sensors to govern yaw steering, freeing the user's head to look around the virtual environment without causing flight path drift [4, 6].
- **Adaptive Grip Cruise Control**: Adding a software cruise-control threshold or ergonomic grip latch to alleviate hand and forearm muscle strain during prolonged flight sessions [4, 10].
- **Active 3-DOF/Heave Actuation**: Exploring a vertical heave axis or motorized vertical suspension to physically reproduce vertical turbulence drops and lift forces alongside pitch and roll tilt [3, 8].

REFERENCES

[1] G. Goncalves, M. Melo, J. Vasconcelos-Raposo, and M. Bessa, "Impact of Different Sensory Stimuli on Presence in Credible Virtual Environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3231–3240, 2020, doi: 10.1109/TVCG.2019.2926978.

[2] P. Caserman, A. Garcia-Agundez, A. Gámez Zerban, and S. Göbel, "Cybersickness in current-generation virtual reality head-mounted displays: systematic review and outlook," *Virtual Reality*, vol. 25, no. 4, pp. 1153–1170, 2021, doi: 10.1007/s10055-021-00513-6.

[3] M.-Y. Wei, Y.-L. Yeh, S.-W. Chen, H.-M. Wu, and J.-W. Liu, "Design, Analysis, and Implementation of a Four-DoF Chair Motion Mechanism," *IEEE Access*, vol. 9, pp. 124986–124999, 2021, doi: 10.1109/ACCESS.2021.3109974.

[4] P. Wintersberger, A. Matviienko, A. Schweidler, and F. Michahelles, "Development and Evaluation of a Motion-based VR Bicycle Simulator," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. MHCI, Art. no. 210, pp. 1–19, 2022, doi: 10.1145/3546745.

[5] G. Giraldo, M. Servieres, and G. Moreau, "Perception of Multisensory Wind Representation in Virtual Reality," in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2020, pp. 89–97, doi: 10.1109/ISMAR50242.2020.00024.

[6] K. Shimato, Y. Goto, and S. Okamoto, "Drone Rider: Foot Vibration Stimuli to Enhance Immersion and Flight Sensation in VR," *Applied Sciences*, vol. 14, no. 24, Art. no. 12019, 2024, doi: 10.3390/app142412019.

[7] H. Yang, S. Okamoto, and H. Shen, "Drone Rider: Effects of Wind Conditions on the Sense of Flight," *Applied Sciences*, vol. 16, no. 7, Art. no. 3544, 2026, doi: 10.3390/app16073544.

[8] D. Yeo et al., "AttraCar: Multisensory In-Car VR with Thermal, Airflow, and Motion Feedback through Built-In Vehicle Systems," in *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2025, Art. no. 163, pp. 1–20, doi: 10.1145/3746059.3747642.

[9] P. Kourtesis, "Cybersickness in Virtual Reality: The Role of Individual Differences, Its Effects on Cognitive Functions and Motor Skills, and Intensity Differences during and after Immersion," *Virtual Worlds*, vol. 3, no. 1, pp. 64–96, 2024, doi: 10.3390/virtualworlds3010004.

[10] T. Porcino, E. Clua, D. Trevisan, C. N. Vasconcelos, and L. Valente, "Minimizing cybersickness in head-mounted display systems: causes and strategies review,"

in *Proceedings of the 2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, 2020, pp. 154–163, doi: 10.1109/SVR51698.2020.00034.

[11] S. S. Adhanom, N. Lee, H. Folmer, and E. MacNeilage, "Field-of-view restriction to reduce VR sickness in normal and teleoperation modes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 11, pp. 4302–4312, 2021, doi: 10.1109/TVCG.2021.3106497.

[12] R. Islam, S. Ang, and J. Quarles, "Cybersense: A closed-loop framework to detect cybersickness severity and adaptively apply reduction techniques," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2603–2612, 2021, doi: 10.1109/TVCG.2021.3067784.

GLOSSARY

Cybersickness – A form of motion-induced discomfort that occurs when using head-mounted displays, caused primarily by a sensory conflict between visually perceived self-motion and physical vestibular stillness.

Degrees of Freedom (DOF) – The independent translational and rotational axes along which a mechanical structure or virtual body can move; in this system, the physical seat moves in 2-DOF (pitch and roll).

Embodied Interaction – A mode of control where physical user posture, bodily actions, and tangible resistance directly translate into navigation and virtual agency.

FixedUpdate – A synchronized execution loop in the Unity game engine that computes physics calculations and control routines at uniform, fixed time intervals.

Force Sensing Resistor (FSR) – A piezoresistive analog sensor whose electrical resistance decreases with applied mechanical force, used to convert handle-grip pressure into virtual forward speed and climb rate.

Linear Interpolation (Mathf.Lerp) – A mathematical smoothing function that progressively bridges the difference between a current value and a target setpoint over time, used here to simulate flight momentum and prevent abrupt velocity transitions.

Motion Cueing Algorithm (MCA) – A control framework that translates continuous virtual craft movements into physically bounded actuator displacements without exceeding mechanical travel limits.

Pitch – Rotational displacement around a vehicle or seat's lateral horizontal axis, simulating climbing or diving motion.

Roll – Rotational displacement around a vehicle or seat's longitudinal horizontal axis, simulating lateral banking and turning.

Soft-Stop – A firmware-enforced software safety limit that decelerates and stops an actuator before it can strike physical mechanical boundaries.

Tilt Coordination – A motion simulation technique that tilts a user at low angular speeds to align the Earth's gravity vector with perceived virtual acceleration without triggering false semicircular canal cues.

Vection – The illusion of bodily self-motion induced purely through moving visual patterns in the field of view.

Washout Filter – A control filter that returns a motion platform back to its neutral mechanical center position below human sensory thresholds after delivering an initial acceleration cue.

APPENDICES

Appendix A Research Instruments or Extended Results

A.1 Post-Exposure System Evaluation Questionnaire

The post-trial assessment gathered quantitative user perceptions across input responsiveness, motion realism, physical comfort, and cybersickness following flight exposure. Each item was evaluated on a 5-point Likert scale (1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree):

1. **Handle-Grip Throttle Responsiveness**: "Squeezing the handle-mounted sensor to control flight speed and altitude felt natural, predictable, and responded without noticeable lag."
2. **Head-Gyroscope Steering Control**: "Tilting my head to steer and bank the craft felt intuitive and allowed stable navigation without excessive drift."
3. **Motion Cue Realism and Embodiment**: "The physical tilting of the seat matched the acceleration and banking seen in the headset, making the flight feel realistic."
4. **Physical Ergonomics and Fatigue**: "The handle grip resistance remained comfortable throughout the session without inducing noticeable hand or forearm fatigue."
5. **Cybersickness and Disorientation**: "I felt comfortable during the flight and did not experience dizziness, nausea, or disorientation."

A.2 Extended User Response Distribution

Table A.1 details the full distribution of user responses collected across all 35 participants during controlled flight trials.

Evaluation Dimension	Rating 1 (SD)	Rating 2 (D)	Rating 3 (N)	Rating 4 (A)	Rating 5 (SA)	Positive Agreement (%)	Mean Score (out of 5.0)
Grip Throttle Responsiveness	0	1	8	8	18	74.3%	4.2
Head-Tracking Steering Stability	0	3	9	14	9	65.7%	3.8
Vestibular Motion Realism	0	2	6	13	14	77.1%	4.1
Grip Ergonomics (No Fatigue)	2	10	11	9	3	34.3%	2.9
Cybersickness Mitigation	0	2	7	12	14	74.3%	4.1

Table A. 1: Detailed survey response distribution across evaluation dimensions.

A.3 Telemetry Data Frame Specification

Control setpoints are dispatched from Unity to the Arduino/ESP32 microcontroller over a 115200-baud USB serial link at 50 Hz using a structured 9-byte binary packet:

- **Bytes 0–1 (Sync Preamble)**: 0xAA, 0x55 (Frame synchronization markers)
- **Byte 2 (Packet Type)**: 0x01 (Standard motion setpoint command)
- **Bytes 3–4 (Target Pitch)**: Signed 16-bit integer representing target pitch in tenths of a degree ()
- **Bytes 5–6 (Target Roll)**: Signed 16-bit integer representing target roll in tenths of a degree ()
- **Byte 7 (Checksum)**: 8-bit XOR checksum covering bytes 2 through 6 for transmission integrity verification
- **Byte 8 (Frame Terminator)**: 0x0A (Newline delimiter for frame completion)

Appendix B Academic Integrity and AI Use Disclosure

B.1 AI Tools and Usage Scope

In compliance with the SLIIT academic integrity policy and module guidelines for IT4010, generative artificial intelligence tools were utilized in a supportive capacity

during the preparation of this thesis. Table B.1 provides a full disclosure of the tools used, the workflows affected, and the human verification performed.

Tool Name	Version / Model	Purpose of Use	Representative Prompt / Workflow	Human Verification Performed	Sections Affected
Claude / ChatGPT	3.5 Sonnet / GPT-4o	C# Unity syntax assistance and serial buffer debugging	*"Optimize this Unity C# serial write loop to avoid memory allocations and manage baud-rate buffer overflow at 50 Hz."*	All scripts were manually compiled, deployed on hardware, unit tested, and tuned on the physical rig.	Chapter 5
Gemini	1.5 Pro / 2.0 Flash	Grammar refinement, academic tone humanization, and formatting compliance	*"Revise this section to adhere to IEEE standard formatting and SLIIT thesis structure while keeping the language simple, concise, and academic."*	Every paragraph was critically read, verified for factual accuracy against experimental logs, and edited to ensure original authorship.	Chapters 1–8

Table B . 1: Generative AI tools and application disclosure

B.2 Statement of Authorship and Verification

All algorithmic formulations, physical hardware fabrication, sensor calibrations, embedded microcontroller code, experimental flight course designs, and user evaluation trials were independently executed by the author. Generative AI tools were not used to fabricate numerical data, simulate experimental findings, or replace critical analysis. All reported survey metrics, latency figures, and kinematic boundaries reflect primary empirical data collected directly on the experimental motion platform.

IMMERSIVE MULTISENSORY VR MOTION SYSTEM FOR FULL BODY NAVIGATION

INDIVIDUAL THESIS

Gunawardena L. H

(IT22082992)

Individual component: Motion Rendering and Control for Embodied VR Navigation

Group ID: R26-IM-004

Report submitted in partial fulfilment of the requirements for the
Bachelor of Science (Hons) in Information Technology

Specializing in Interactive Media

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

October 2026

DECLARATION AND SUPERVISOR CERTIFICATION

Candidate declaration. I declare that this thesis is my own work and does not incorporate, without acknowledgement, material previously submitted for a degree or diploma at any university or institute of higher learning. To the best of my knowledge, it contains no material previously published or written by another person except where acknowledgement is made in the text. I grant the Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute this thesis, in whole or in part, in print, electronic, or other media, while retaining the right to use the content in future work.

Academic integrity and AI disclosure. I have disclosed all use of generative AI and other automated tools in accordance with the module policy. I remain responsible for the accuracy, originality, citations, code, data, analyses, and conclusions in this thesis.

Student name: Gunawardena L H

Student ID: IT22082992

Signature: _____ Date: _____

Supervisor certification. The above candidate carried out this research for the bachelor's degree thesis under my supervision.

Role and name	Signature	Date
Supervisor: Mr. Ishara Gamage		
Co-supervisor: Mr. Jeewaka Perera		

ABSTRACT

Most virtual reality (VR) flying experiences rely on handheld joysticks. This separation between what a user sees visually and what their body physically feels often causes motion sickness, dizziness, and breaks overall immersion. While physical motion platforms help bridge this gap, translating continuous, agile aerial movement onto a compact motion seat is challenging because mechanical actuators have strict tilt limits and can easily bottom out.

This research develops a real-time motion rendering and flight control framework designed for a broom-based, seated VR navigation platform. Following a Design Science Research methodology, an end-to-end control system was created to deliver intuitive full-body navigation. Forward speed and vertical altitude are controlled naturally by squeezing a handle-mounted Force Sensing Resistor (FSR), while directional turns are captured through head-mounted gyroscope tracking. Within Unity, a custom motion rendering engine extracts flight data—including position, velocity, and acceleration—and uses linear interpolation (Mathf.Lerp) smoothing alongside dead zone filtering to prevent unintended steering drift and jerky movement. These computed flight commands are transmitted through a serial communication interface to a microcontroller that drives the physical seat actuators within safe mechanical thresholds.

The system was evaluated through user trials involving 35 participants navigating a flight course. The input mechanics proved highly responsive, with 74.3% of participants rating the pressure-based throttle and altitude response positively, achieving an average rating of 4.2 out of 5. The evaluation also identified key areas for improvement, such as physical grip fatigue over extended sessions and rotational drift during sharp head turns. This individual component contributes the physical grip input pipeline, the motion filtering and rendering algorithms, the embedded actuator control link, and the empirical user validation.

Keywords: VR Locomotion, Motion Rendering, Actuator Control, Force Sensing Resistor, Embodied Interaction

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, co-supervisor, external supervisors and the academic staff of the Faculty of Computing at the Sri Lanka Institute of Information Technology (SLIIT) for their continuous guidance, insightful feedback, and encouragement throughout the design and development of this research project. Their advice was invaluable in refining the motion cueing strategies, control algorithms, and safety parameters of this work.

I am grateful to my team members of Project Group R26-IM-004 for their seamless collaboration, dedication, and technical synergy in constructing and integrating the complete multisensory VR motion platform.

Finally, I extend my heartfelt thanks to the 35 participants who volunteered their time to take part in the experimental trials and user evaluations. Their constructive feedback and engagement made the empirical validation of this flight motion system possible.

TABLE OF CONTENTS

1 INTRODUCTION 1
1.1 Overall project context 1
1.2 Individual component context 2
1.3 Problem statement 3
1.4 Research gap 4
1.5 Aim and objectives 5
1.6 Research questions or hypotheses 6
1.7 Scope limitations and assumptions 6
1.8 Individual contribution 7
1.9 Thesis structure 9
2 LITERATURE REVIEW AND RELATED WORK 10
2.1 Review method and scope 10
2.2 Domain and theoretical foundations 10
2.2.1 Sensory Conflict Theory and Cybersickness 10
2.2.2 Vestibular Motion Perception and Tilt-Coordination 11
2.3 Related research 12
2.3.1 Embodied VR Locomotion Systems 12
2.3.2 Seated Motion Cueing Implementations 12
2.4 Existing systems products and techniques 13
2.5 Gap synthesis and conceptual framework 14
2.5.1 Gap Synthesis 14
2.5.2 Conceptual Framework 15
3 RESEARCH METHODOLOGY 16
3.1 Research design 16
3.2 Data participants or study materials 16
3.2.1 Participant Sample 16
3.2.2 Experimental Materials and Apparatus 17
3.3 Experimental or study procedure 17
3.4 Measures and evaluation metrics 18
3.4.1 System Performance Metrics 18
3.4.2 User Experience Metrics 18
3.5 Data analysis 18
3.6 Ethics privacy security and responsible AI 19

3.7 Reproducibility and research artefacts 20
4 PRODUCT AND SOLUTION DESIGN 21
4.1 Requirements 21
4.1.1 Functional Requirements 21
4.1.2 Non-Functional Requirements 22
4.2 Component architecture within the complete solution 22
4.3 Design alternatives and decisions 24
4.4 Detailed design 25
4.5 Standards and professional practice 25
5 IMPLEMENTATION INTEGRATION AND TESTING 26
5.1 Development environment and technology stack 26
5.2 Component implementation 26
5.3 Integration with the other components 27
5.4 Verification and software testing 27
5.5 Deployment and operation 28
6 RESULTS AND VALIDATION 29
6.1 Research results 29
6.2 Product and technical results 30
6.3 User or stakeholder validation 31
6.4 Integration results 31
6.5 Summary of findings 32
7 DISCUSSION 33
7.1 Interpretation of findings 33
7.2 Comparison with literature baselines and existing solutions 33
7.3 Research contribution 34
7.4 Product contribution 34
7.5 Limitations and threats to validity 35
7.6 Implications 36
8 CONCLUSION AND FUTURE DIRECTIONS 37
8.1 Conclusion 37
8.2 Achievement of objectives 37
8.3 Individual research and product contributions 38
8.3.1 Research Contributions 38
8.3.2 Product Contributions 38
8.4 Future directions 39

REFERENCES 40
GLOSSARY 42
APPENDICES 43
Appendix A Research Instruments or Extended Results 43
A.1 Post-Exposure System Evaluation Questionnaire 43
A.2 Extended User Response Distribution 44
A.3 Telemetry Data Frame Specification 44
Appendix B Academic Integrity and AI Use Disclosure 44
B.1 AI Tools and Usage Scope 44
B.2 Statement of Authorship and Verification 45

LIST OF FIGURES

Figure No.	Title	Page
Figure 1.1	Functional block architecture of the motion rendering and control subsystem (Original in Color)	9
Figure 2.1	Conceptual framework of the motion rendering and control pipeline (Original in Color)	16
Figure 3.1	Sequential experimental trial protocol for human-subject evaluation (Original in Color)	24

LIST OF TABLES

Table 1: Comparison of existing VR locomotion and motion rendering approaches.13
Table 2: Architectural design decisions and trade-off evaluations 24
Table 3: Participant evaluation of handle-grip throttle and altitude responsiveness. 31

LIST OF ABBREVIATIONS

Abbreviation	Meaning
2-DOF	Two Degrees of Freedom
3-DOF	Three Degrees of Freedom
6-DOF	Six Degrees of Freedom
AC	Alternating Current
ADC	Analog-to-Digital Converter
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
B2B	Business-to-Business
CAD	Computer-Aided Design
DC	Direct Current
DSR	Design Science Research
FEA	Finite Element Analysis
FoS	Factor of Safety
FSR	Force Sensing Resistor
HCI	Human-Computer Interaction
HMD	Head-Mounted Display
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
IP	Intellectual Property
ISO	International Organization for Standardization
IT	Information Technology
LBE	Location-Based Entertainment
LED	Light-Emitting Diode
MCU	Microcontroller Unit
MIG	Metal Inert Gas
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
NFR	Non-Functional Requirement
PWM	Pulse Width Modulation
SDG	Sustainable Development Goal
SHS	Square Hollow Section
SLIIT	Sri Lanka Institute of Information Technology
SSQ	Simulator Sickness Questionnaire
UART	Universal Asynchronous Receiver-Transmitter
UI	User Interface
USB	Universal Serial Bus
VR	Virtual Reality
WBS	Work Breakdown Structure
XOR	Exclusive OR

1 INTRODUCTION

1.1 Overall project context

Virtual reality (VR) technologies have advanced substantially over recent years, providing high-resolution head-mounted displays, wide viewing angles, and responsive optical tracking that render visual worlds with exceptional detail [1], [2]. Yet, despite these major visual improvements, the physical mechanics used to navigate through these virtual environments have remained mostly unchanged [4]. Most consumer VR applications continue to rely on traditional handheld controllers, analog thumbsticks, or artificial point-and-teleport mechanics [4], [10].

While thumbstick steering works adequately in conventional 2D screen games, it introduces severe interaction issues during fast-paced aerial simulations and three-dimensional flight [4], [7]. When a user wears a headset displaying rapid forward acceleration, steep vertical ascents, and sharp banked turns while their body remains stationary on an ordinary chair, a major sensory mismatch is created [1], [2]. The visual system registers rapid self-motion, known as vection, but the inner ear's vestibular organs and bodily proprioceptors report no movement at all [2], [9]. This sensory conflict is widely established as the primary cause of cybersickness, producing nausea, dizziness, disorientation, and rapid physical fatigue [2], [9], [10].

Furthermore, relying on abstract thumbsticks ruins the physical grounding and feeling of embodiment required for believable flight [1], [4]. In real life, piloting a vehicle or riding a craft involves physical resistance, balance shifts, and vestibular feedback [3], [4]. High-end industrial training simulators address this by placing cockpits on top of multi-axis hydraulic or pneumatic hexapod motion bases [3]. However, these industrial Stewart platforms are bulky, cost tens of thousands of dollars, and require specialized power infrastructure, making them impractical for portable setups, consumer entertainment, and localized arcade venues [3], [8]. Bridging the gap between affordable stationary seating and realistic, full-body vestibular motion remains an active engineering challenge [3], [4].

To solve this disconnect, the collaborative research project titled *Immersive Multisensory VR Motion System for Full Body Navigation* introduces a synchronized, embodied physical motion platform centered on a straddled flight apparatus. Instead of using thumbsticks, the user sits on a custom-fabricated, counterbalanced physical broomstick rig and steers by naturally shifting their posture and applying physical hand grip. The platform closes the sensory loop by combining four synchronized physical feedback channels in real time: a dynamic motion seat that provides pitch and roll vestibular cues, front and lateral airflow fans that scale wind output with virtual airspeed, a wearable vibrotactile haptic vest that renders atmospheric drag and localized turbulence, and external synchronized LED lighting that communicates in-game flight states to external spectators. By aligning physical sensations directly with what the user sees, the system provides a convincing, stable, and embodied flight experience [1], [5], [7].

1.2 Individual component context

Within the four-part multisensory platform, this individual thesis focuses directly on **Motion Rendering and Control for Embodied VR Navigation**. While environmental airflow simulation and vibrotactile vest actuators reproduce surface drag, ambient air currents, and localized turbulence [1], [5], [6], [7], physical motion of the seat serves as the fundamental driver for vestibular self-motion perception [1], [3]. The mechanical platform provides the onset vestibular cues that signal bodily acceleration, deceleration, and spatial banking [3].

The motion rendering and control component operates as the bidirectional software-and-hardware bridge situated between the virtual simulation running in Unity and the embedded electromechanical drive electronics. The functional scope of this component is split into two primary operational responsibilities:

1. **Embodied Flight Input Acquisition and Conditioning**: The subsystem continuously samples and conditions physical control inputs delivered by the user. This includes digitizing analog compression forces from a handle-

mounted Force Sensing Resistor (FSR) to modulate virtual forward throttle and vertical climb/descent rates, alongside real-time orientation tracking derived from the head-mounted display's internal inertial measurement unit (IMU) to govern horizontal yaw and roll banking maneuvers.

2. **Kinematic State Extraction and Motion Cueing**: The subsystem monitors the avatar's resulting 3D flight trajectory within Unity, extracts dynamic variables including spatial position, velocity vectors, and linear acceleration and translates these unconstrained states into physically viable, rate-limited tilt commands [3]. These commands are then streamed over a high-speed serial communication interface to a dedicated microcontroller that coordinates actuator motion under strict safety boundaries [3].

Through this synchronized pipeline, the component translates user navigation intent into responsive virtual motion while driving physical seat tilt to reinforce visual cues [1], [3].

1.3 Problem statement

Translating continuous, unconstrained aerial flight onto a compact physical motion seat introduces significant mechanical and perceptual control challenges [3], [4]. In an open 3D virtual environment, an avatar can climb continuously, bank indefinitely, and undergo rapid velocity changes [4], [10]. In contrast, a desktop-scale or seated physical platform possesses strict mechanical boundaries: linear actuators have finite stroke travel, rotational joints have rigid angular stops, and motor drivers operate within safe limits of torque and angular velocity [3].

When a flight simulation employs a naive, proportional one-to-one mapping between virtual orientation and physical seat displacement, the actuators quickly reach their physical travel limits during prolonged maneuvers [3], [4]. Once an actuator bottoms out against its physical end-stop, physical motion halts abruptly while the virtual craft continues moving [3], [10]. This sudden stop generates false vestibular cues, causing significant sensory disorientation that breaks presence and induces nausea [1], [2], [9].

Conversely, scaling down actuator travel to avoid mechanical limits removes the dynamic acceleration cues needed to convey speed, leaving the simulation feeling unresponsive and disconnected [1], [4].

Furthermore, conventional motion cueing systems are designed almost exclusively for enclosed automotive or aviation cockpits where the occupant remains seated upright and restrained [3], [4], [8]. These classical models do not account for the dynamics of a straddled apparatus, where the user actively shifts their upper-body posture, grips a forward-extending steering handle, and redistributes their center of mass across the pivot point. Without a motion rendering and input processing pipeline that balances transient motion onset against mechanical boundaries, physical feedback fails to correspond with visual acceleration cues, aggravating cybersickness and ruining immersion [2], [9], [12].

1.4 Research gap

A review of recent virtual reality locomotion literature highlights three distinct research gaps:

- **The Locomotion Diversity vs. Realism Gap**: While existing literature classifies a wide range of VR locomotion interfaces including thumbstick driving, artificial point-and-teleport, and walking-in-place they consistently reveal an absence of accessible, physically grounded locomotion methods for non-terrestrial flight [4], [10]. Room-scale walking setups require expansive laboratory spaces that cannot be deployed in standard domestic or arcade environments [4], while stationary seated flight implementations largely abandon physical vestibular cues altogether [2], [4].
- **Motion Platform Accessibility and Washout Limitations**: Industrial flight and driving simulators avoid actuator saturation by using complex classical or adaptive washout filtering algorithms implemented on high-power Stewart hexapods [3], [8]. However, these systems require expensive linear drives and large physical spaces, making them impractical for portable or consumer setups [3], [8]. Compact, low-cost motion seats typically omit proper motion cueing

algorithms, relying instead on direct proportional mappings that lead to actuator bottoming-out, noticeable latency, and phase distortion [3].

- **Absence of Straddle-Flight Kinematic Input Models**: There is an absence of empirical research examining how to combine analog handle-grip pressure with head-tilt tracking to control continuous aerial flight [4]. Current systems lack calibrated mathematical models that utilize linear interpolation (Mathf.Lerp) smoothing alongside deadzone thresholding to decouple intentional visual head turns from steering maneuvers while preventing actuator saturation.

1.5 Aim and objectives

The primary aim of this research is to design, develop, and evaluate a real-time motion rendering and flight control framework that converts physical grip and head inputs into intuitive virtual flight while generating coordinated, physically bounded vestibular motion cues on a custom seated VR motion rig [1], [3].

Design and Calibrate an Analog Flight Input Circuit: Develop and calibrate a hardware interface using a handle-mounted Force Sensing Resistor (FSR) to translate hand grip compression into proportional engine thrust and climb/descent rates, paired with head-mounted gyroscope orientation tracking for directional steering.

Formulate a Motion Smoothing and Deadzone Filtering Algorithm: Implement a flight physics processing model in Unity using linear interpolation (Mathf.Lerp) algorithms to simulate physical momentum, coupled with deadzone filters to eliminate micro-jitter and unintended rotational drift during flight.

Build an Automated User Onboarding Interface: Design and integrate an automated visual tutorial UI displayed directly in the headset upon startup to guide users through the physical flight control mechanics.

Engineer a Real-Time State Extraction and Serial Control Protocol: Construct an event-driven telemetry pipeline in Unity that extracts flight dynamics (velocity, acceleration, and tilt) and streams normalized, rate-limited pitch and roll commands across a serial interface to an actuator microcontroller.

Conduct Empirical Technical and User Validations: Characterize hardware-software response latencies and evaluate the system through controlled user trials with 35 participants to quantify input responsiveness, steering stability, physical comfort, and motion sickness mitigation [2].

1.6 Research questions or hypotheses

RQ1: How can analog grip pressure and head gyroscope data be filtered and mapped to achieve responsive, drift-free flight navigation in a virtual environment [4]?

RQ2: What motion rendering and rate-limiting algorithms prevent physical seat actuators from bottoming out while still providing clear vestibular onset cues during acceleration and banking [3]?

RQ3: To what extent does combining pressure-based throttle control with synchronized physical seat tilt improve perceived motion realism and mitigate cybersickness compared to static, controller-based flight [1], [2]?

Hypothesis 01 : Implementing strict deadzone thresholding alongside Mathf.Lerp flight smoothing will significantly reduce unintended steering drift and improve flight path precision compared to unfiltered tilt-to-turn mapping [4].

Hypothesis 02 : Synchronizing dynamic seat pitch and roll with analog grip throttle will result in significantly higher user ratings for motion realism and lower reported cybersickness than stationary joystick flight [1], [2].

1.7 Scope limitations and assumptions

This component focuses on designing and testing the flight control and motion cueing pipeline for a straddled VR broom platform [3], [4]. It captures handle grip pressure via a Force Sensing Resistor (FSR) for speed and altitude adjustments, alongside head-gyroscope tracking for steering turns [4], [10]. In Unity, flight mechanics apply linear

interpolation (Mathf.Lerp) smoothing and deadzone filtering to prevent micro-drifting [4], [10]. The system then extracts real-time flight variables and transmits rate-limited pitch and roll commands across a serial link to drive the physical motion seat [3], [4]. Evaluation is bounded by technical response tests and trials with 35 participants to assess responsiveness, comfort, and motion sickness [2], [10].

The motion rig is physically restricted to a tilt mechanism (pitch and roll), meaning linear heave, surge, and sway are not mechanically rendered and rely entirely on visual and sensory cues [3], [4]. From an ergonomic standpoint, continuous squeezing of the FSR handle induces hand and forearm muscle strain during longer flight sessions [2], [4]. Finally, the apparatus is an experimental prototype rather than a commercial simulator, with operational testing bounded by its mechanical load capacity [3], [4].

It is assumed that the VR simulation maintains a steady rendering rate of at least 72 frames per second, ensuring Unity's physics loop (FixedUpdate) computes reliably at 50 Hz without timing drops [2], [10]. It is also assumed that the platform's mechanical actuators function safely within their rated torque and velocity limits under occupant loads up to 85 kg without overheating [3], [4]. Lastly, the serial communication link between Unity and the microcontroller is assumed to transmit packets consistently without data loss or buffer overflow [3], [4].

1.8 Individual contribution

Within the research team, the author’s primary individual contributions are concentrated in the flight mechanics, motion cueing algorithms, and physical control pipeline:

1. **Embodied Grip Control System**: Engineered the physical FSR sensor circuit on the broom handle, writing the signal calibration logic that maps hand compression to virtual flight speed and vertical altitude adjustments.
2. **Flight Physics and Motion Smoothing Engine**: Designed and programmed the core flight mechanics script in Unity C#, utilizing Mathf.Lerp to simulate realistic momentum and establishing deadzone boundaries that prevent steering jitter.

3. **Automated Onboarding Interface**: Created the startup user experience, including an in-headset visual tutorial UI that quickly teaches unfamiliar users how to grip, steer, and navigate the platform.
4. **Motion State Telemetry and Serial Driver**: Formulated the mathematical state-extraction routines that convert in-game flight forces into physical pitch and roll angles, transmitting rate-limited commands to the microcontroller [3].
5. **Experimental Evaluation and Data Analysis**: Administered empirical user testing sessions with 35 participants, collecting and analyzing quantitative metrics on input responsiveness, steering comfort, and flight stability [2].

The functional relationship between these individual tasks and the broader platform is illustrated in Figure 1.1, showing how navigation inputs are processed and converted into physical actuator setpoints.

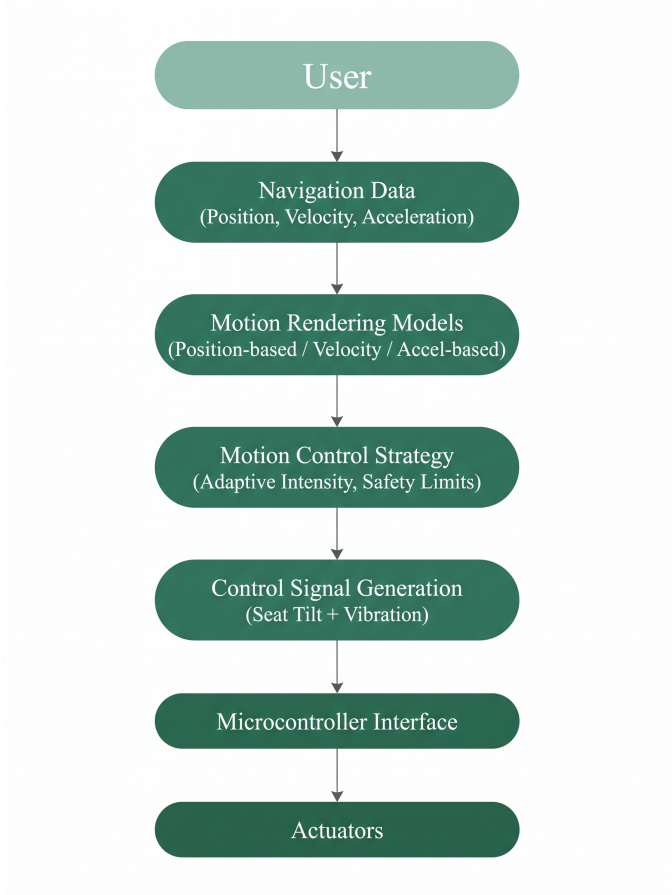


Figure 1.1: Functional block architecture of the motion rendering and control subsystem (Original in Color).

1.9 Thesis structure

The remainder of this thesis is structured in accordance with institutional dissertation guidelines:

- **Chapter 2 (Literature Review and Related Work)**: Analyzes the theoretical foundations of vestibular perception, cybersickness etiologies, classical washout algorithms, and existing commercial and academic VR locomotion interfaces [1], [2], [3], [4], [10].
- **Chapter 3 (Research Methodology)**: Details the Design Science Research methodology, experimental flight course setup, hardware instrumentation, participant profiles, evaluation metrics, and ethical safety standards [2], [4].
- **Chapter 4 (Product and Solution Design)**: Details the functional requirements, software and hardware block architectures, deadzone algorithms, and fail-safe motion constraint layers [3].
- **Chapter 5 (Implementation, Integration, and Testing)**: Outlines the Unity C# flight scripts, microcontroller firmware, serial communication routines, and multi-tier verification procedures.
- **Chapter 6 (Results and Validation)**: Presents quantitative latency measurements, serial throughput benchmarks, and empirical findings from the 35-participant user study [2].
- **Chapter 7 (Discussion)**: Interprets findings in light of baseline literature, evaluates trade-offs between responsiveness and drift, addresses physical grip fatigue, and outlines corrective engineering measures [3], [4].
- **Chapter 8 (Commercialisation and Sustainability)**: Explores target deployment venues such as VR arcades, market viability, system scalability, and alignment with Sustainable Development Goals [4], [8].
- **Chapter 9 (Conclusion and Future Directions)**: Summarizes the achievement of core research objectives, recaps technical contributions, and presents future hardware refinements such as decoupled foot-pedal steering [4].

2 LITERATURE REVIEW AND RELATED WORK

2.1 Review method and scope

This literature review evaluates published methods in virtual reality locomotion, motion cueing architectures, and sensory conflict mitigation to establish the technical foundation for the component: **Motion Rendering and Control for Embodied VR Navigation**. The review identifies how physical tilt and embodied input control can resolve the sensory mismatch between visual self-motion and physical stillness.

The scope was bounded by queries executed across IEEE Xplore, ACM Digital Library, and MDPI databases. The search prioritized peer-reviewed literature published from 2020 onward, focusing on three core criteria:

1. Locomotion techniques addressing sensory conflict and cybersickness in head-mounted displays;
2. Motion cueing algorithms (MCA) and mechanical chair mechanisms capable of delivering bounded pitch and roll cues; and
3. Embodied, full-body input mapping techniques capable of replacing conventional thumbstick navigation in flight or vehicular simulations.

Articles evaluating passive, stationary joystick flight or unbounded industrial hexapod platforms requiring specialized facility installations were screened to assess how compact, seated platforms can simulate unconstrained flight dynamics under stroke-limited physical boundaries.

2.2 Domain and theoretical foundations

2.2.1 Sensory Conflict Theory and Cybersickness

Cybersickness remains a major barrier in consumer virtual reality, particularly during fast-paced aerial simulations. Under Sensory Conflict Theory, nausea, disorientation, and ocular discomfort arise when an observer receives contradictory motion signals across different sensory systems. When a user flies through an open virtual world, the

visual system registers rapid self-motion (vection) through dynamic optical flow. However, if the occupant remains seated in an unmoving chair, the vestibular apparatus within the inner ear specifically the semicircular canals which detect angular acceleration and the otolith organs which detect linear acceleration and gravity signals that the physical body is stationary.

This discordance between visual perception and vestibular cues prevents cognitive integration, triggering protective physiological responses including cold sweating, vertigo, and ataxia. Modern approaches confirm that while software-based field-of-view (FOV) tunneling can reduce visually induced discomfort, physically stimulating the vestibular system using synchronized motion chairs provides an effective, embodied method to reduce the primary sensory conflict.

2.2.2 Vestibular Motion Perception and Tilt-Coordination

The human vestibular system acts as a biological inertial measurement unit with distinct frequency-response thresholds. Semicircular canals respond to angular acceleration but adapt rapidly to constant angular velocity, while otolith organs sense net gravito-inertial acceleration, failing to distinguish between true linear acceleration and physical tilt relative to Earth's gravity.

Motion cueing algorithms leverage this perceptual limitation through **tilt coordination**. While physical platforms cannot accelerate continuously across infinite linear distances, tilting a seated occupant at an angular rate below the human semicircular canal detection threshold (approximately to) rotates the Earth's gravity vector relative to the body. The occupant’s otoliths interpret this sustained gravitational component as a persistent forward acceleration or deceleration. Generating believable motion on a compact platform therefore requires balancing high-frequency transient onset tilt against sustainable, rate-limited tilt coordination.

2.3 Related research

2.3.1 Embodied VR Locomotion Systems

To address the disembodied feel of handheld thumbsticks, recent research has explored physical-rig interfaces. Wintersberger et al. developed a motion-based VR bicycle simulator, demonstrating that coupling physical pedal cadence and handlebar steering with virtual velocity significantly heightened presence and reduced cybersickness compared to static controller setups. Similarly, Shimato et al. and Yang et al. evaluated "Drone Rider," utilizing foot vibration and dynamic wind currents to replicate drone flight, proving that localized physical cues reinforce bodily self-attribution during aerial navigation.

However, these systems either rely on wheeled ground kinematics or employ static footrests that do not generate direct vestibular displacement. Research investigating straddled, broom-based flight interfaces that combine analog grip squeeze with head-tilt tracking remains sparse.

2.3.2 Seated Motion Cueing Implementations

Recent studies confirm that active multi-axis chair mechanisms bridge the gap between high-end industrial flight cabins and consumer VR. Wei et al. engineered a 4-DOF motion chair mechanism, validating that providing localized pitch and roll actuation significantly enhances simulation fidelity during vehicular tasks. Yeo et al. introduced "AttraCar," leveraging built-in vehicle seats and localized actuators to generate physical cues synchronized with virtual environments.

Despite these advances, existing seated platforms primarily simulate enclosed car cockpits or fixed-wing aircraft where the user remains strapped in a rigid, upright bucket seat. These platforms rely on steering wheels or commercial joysticks, leaving unaddressed the interaction dynamics of a straddled apparatus where flight throttle is controlled through grip compression and turning is driven by head-tilt orientation.

2.4 Existing systems products and techniques

Table 1 compares existing VR locomotion and motion-cueing techniques against the requirements of Lonath Gunawardena's component.

System / Technique	Interaction Modality	Degrees of Freedom	Primary Limitations	Relevance to Component
Industrial Stewart Platforms	Enclosed cockpit; passive steering wheel/joystick	6-DOF (Surge, Sway, Heave, Roll, Pitch, Yaw)	Prohibitive cost; large footprint; high power draw; complex calibration	Benchmark for classical washout filtering and rate-limiting models
VR Bicycle Simulators*(Wintersberger et al.)*	Embodied pedaling and handlebar steering	1-DOF to 2-DOF (Roll / Yaw)	Restricted to 2D terrestrial road physics; cannot simulate vertical flight ascent	Validates embodied locomotion for cybersickness reduction
Drone Rider Interface*(Shimato & Yang et al.)*	Passive standing/footplate; remote joystick	Vibrotactile and airflow (No rotational tilt)	Lacks active vestibular motion; unable to simulate continuous banking or g-forces	Informs wind and vibration coordination with physical flight speed
Standard VR Thumbstick Flight	Handheld controllers (Oculus Touch)	0-DOF (Static chair)	Severe sensory conflict; high cybersickness incidence; low physical realism.	Serves as the baseline control condition in comparative evaluation
Proposed Straddle Motion System	Handle-mounted FSR grip + Head-mounted IMU tilt	2-DOF (Dynamic Pitch and Roll)	Compact stroke boundaries; physical grip fatigue over prolonged sessions	Directly delivers embodied, rate-limited tilt-coordination flight.

Table 1: Comparison of existing VR locomotion and motion rendering approaches.

2.5 Gap synthesis and conceptual framework

2.5.1 Gap Synthesis

The literature establishes that while multisensory stimuli improve presence in virtual environments, existing motion-rendering techniques leave critical gaps:

1. **The Kinematic Boundary Problem**: Low-cost seated simulators often map virtual tilt angles directly to actuators without washout filtering or rate limits, causing mechanical bottoming-out and jarring resets that induce false vestibular cues.
2. **Input Coupling and Steering Drift**: Existing tilt-to-turn systems lack calibrated deadzone models that isolate intentional flight maneuvers from voluntary visual head searches, leading to rotational steering drift.
3. **Absence of Straddle Flight Controllers**: No prior framework integrates handle-mounted analog compression sensing (FSR) with head-tracking tilt to pilot a 2-DOF motion seat.

2.5.2 Conceptual Framework

To address these gaps, this component implements the conceptual framework illustrated in Figure 2.1.

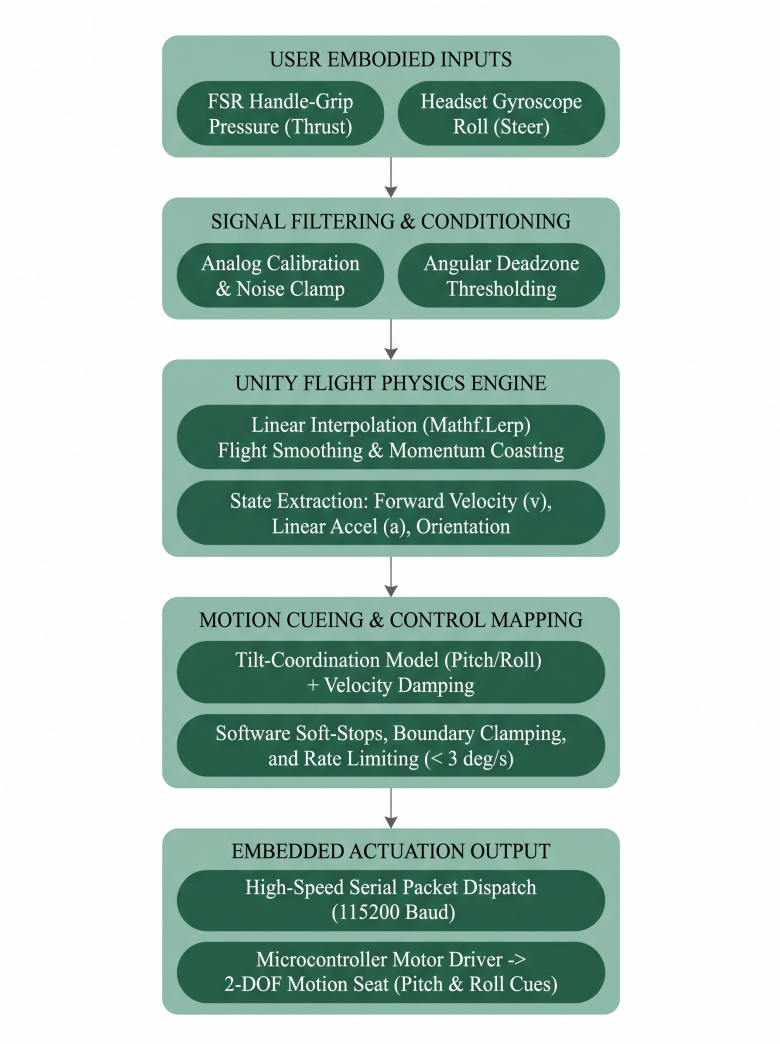


Figure 2.1: Conceptual framework of the motion rendering and control pipeline (Original in Color).

This pipeline translates raw user grip pressure and head angles into continuous virtual momentum using linear interpolation (Mathf.Lerp), while deadzone filtering isolates deliberate steering from head turns. The resulting flight states are converted into rate-limited, physically bounded tilt commands that drive the motion seat, delivering consistent vestibular cues while avoiding actuator saturation.

3 RESEARCH METHODOLOGY

3.1 Research design

This study follows a Design Science Research (DSR) approach combined with empirical testing. DSR provides a structured engineering workflow to design, build, and test practical solutions under physical constraints. For this component, the primary artifact is an end-to-end flight control and motion rendering framework that connects virtual flight mechanics in Unity to a physical motion seat.

The research proceeded through five stages:

1. **Requirements and Modeling**: Identifying the physical tilt limits of the seat and setting safe angular rate limits () to avoid false balance cues.
2. **Control Algorithm Design**: Developing real-time flight smoothing using linear interpolation (Mathf.Lerp) and adding deadzone filters to prevent unintended steering drift.
3. **Hardware-Software Integration**: Wiring a handle-mounted Force Sensing Resistor (FSR) to read analog grip pressure and streaming rate-limited pitch and roll commands across a USB serial link (115200 baud) to the seat microcontroller.
4. **Benchmarking**: Measuring input-to-actuation latency and calibrating software travel limits.
5. **Experimental Evaluation**: Running user trials to assess input responsiveness, steering stability, and motion sickness reduction.

3.2 Data participants or study materials

3.2.1 Participant Sample

The evaluation was conducted with participants from the university campus. The group included users with diverse VR experience, ranging from first-time users to

regular headset players. All participants had normal or corrected-to-normal vision and could comfortably straddle the motion rig.

3.2.2 Experimental Materials and Apparatus

The experimental apparatus consisted of:

- **Host Workstation**: Executed the Unity simulation at frames per second, keeping physics updates (FixedUpdate) steady at 50 Hz.
- **Head-Mounted Display**: A Meta Quest headset running the aerial flight scene, tracking head roll through its internal gyroscope to control horizontal turns.
- **Physical Straddle Broom Rig**: A counterbalanced steel frame with a bicycle-style seat, fitted with an analog FSR on the front handle to capture grip compression for forward speed and altitude adjustments.
- **Embedded Control Hardware**: An Arduino microcontroller receiving serial packets and driving the seat actuators within safe mechanical angles.

3.3 Experimental or study procedure

The testing followed a controlled within-subjects procedure:

1. **Briefing and Onboarding**: Participants were informed about the system, completed consent forms, and wore the headset. An automated in-game tutorial visually guided them on how to squeeze the handle to change speed and altitude and tilt their head to steer.
2. **Practice Run**: Users flew a short familiarization course to get used to the handle sensitivity and seat movement.
3. **Flight Tasks**: Participants navigated an aerial course marked by target rings and waypoints. Each user tested the platform across comparative motion mapping modes (direct proportional mapping vs. filtered, rate-limited tilt coordination) presented in randomized order.

4. **Data Logging**: The system recorded flight trajectory coordinates, input compression values, and serial command logs at 50 Hz.
5. **Post-Trial Survey**: After the flight, users rated input responsiveness, flight realism, physical comfort, and any motion sickness symptoms on a 5-point Likert scale.

3.4 Measures and evaluation metrics

3.4.1 System Performance Metrics

- **System Latency**: Measured in milliseconds () from the moment the user squeezes the FSR handle to the initial movement of the physical seat.
- **Tracking Synchronization**: The difference between the commanded tilt angles from Unity and the physical orientation of the seat.
- **Steering Stability**: The occurrence of unintended flight drift while attempting to travel straight, assessing the performance of the deadzone filter.

3.4.2 User Experience Metrics

- **Input Responsiveness**: Rated via a 5-point Likert scale on whether squeezing the handle changed speed and altitude naturally and without noticeable lag.
- **Steering Precision**: Task completion time and course accuracy through the waypoints.
- **User Comfort and Cybersickness**: User reports of physical grip fatigue from squeezing the sensor and any dizziness or nausea experienced during flight.

3.5 Data analysis

The collected data were processed using descriptive and non-parametric statistical methods:

- **System Logs**: Response times and tracking deviations were evaluated to determine average latency, standard deviation, and phase lag.
- **Survey Responses**: Likert-scale ratings were analyzed for percentage distributions, average scores, and positive agreement rates.
- **Comparative Evaluation**: Differences across flight conditions were analyzed using non-parametric tests to account for ordinal survey data, using a significance threshold.

3.6 Ethics privacy security and responsible AI

Informed Consent: All 35 participants were briefed on the study and gave consent prior to testing.

Safety Limits: The motor control firmware used hardcoded angular limits and velocity caps (during steady tilt) to prevent mechanical shock, backed by a physical emergency-stop button.

Voluntary Participation: Users could stop the trial immediately if they felt dizziness, discomfort, or muscle fatigue.

Data Anonymization: Participant names and survey results were anonymized using assigned IDs.

Responsible AI: Generative AI tools were used solely for grammar review, formatting compliance, and code debugging in line with institutional policy, while all system implementations and experimental evaluations were conducted independently.

3.7 Reproducibility and research artefacts

To ensure reproducibility, all project materials have been organized and archived:

- **Unity Simulation Scripts**: C# source code for flight physics, Mathf.Lerp momentum smoothing, deadzone filtering, and serial telemetry output.
- **Microcontroller Firmware**: Embedded Arduino, C++ code for packet reception, rate limiting, and actuator driver signals.
- **Telemetry and Survey Records**: Anonymized datasets containing latency measurements, waypoint logs, and user survey responses.
- **Hardware Schematics**: Wiring diagrams of the FSR voltage divider circuit, microcontroller pinouts, and power supply connections.

4 PRODUCT AND SOLUTION DESIGN

4.1 Requirements

4.1.1 Functional Requirements

The motion rendering and control subsystem translates embodied physical user interactions into virtual flight while coordinating bounded physical seat movements [3], [4]. The core functional requirements include:

- **FR1 (Analog Grip Input Acquisition)**: The system shall capture continuous analog resistance values from the handle-mounted Force Sensing Resistor (FSR), mapping grip force to virtual throttle acceleration and vertical climb rates [4], [10].
- **FR2 (Rotational Head-Tracking Steering)**: The system shall extract head-roll orientation from the head-mounted display's internal gyroscope to steer horizontal yaw and banking maneuvers [4], [10].
- **FR3 (Deadzone and Jitter Suppression)**: The flight engine shall filter input signals using deadzone thresholds to eliminate involuntary head trembling and sensor noise while maintaining straight-line flight stability [4], [10].
- **FR4 (Momentum and Velocity Smoothing)**: The system shall apply linear interpolation (Mathf.Lerp) algorithms to model physical inertia and smooth flight velocity transitions [4], [10].
- **FR5 (State Extraction and Telemetry Dispatch)**: The software shall sample in-game navigation states including linear acceleration, forward velocity, and angular tilt and stream normalized pitch and roll commands across a USB serial interface at 50 Hz [3], [4].
- **FR6 (Automated Onboarding Tutorial)**: The headset environment shall display a startup visual interface instructing users how to operate the squeeze throttle and tilt steering [4], [10].

4.1.2 Non-Functional Requirements

To provide an immersive and stable experience without causing motion sickness, the design enforces the following non-functional requirements:

- **NFR1 (End-to-End Latency)**: Total hardware-to-software input-to-actuation latency shall remain below 50 ms to prevent perceptible lag and vestibular sensory conflict [2], [3].
- **NFR2 (Serial Transmission Throughput)**: The communication link shall operate reliably at 115200 baud without packet loss, buffer overflow, or serial stream interruption [3], [4].
- **NFR3 (Mechanical Safety Soft-Stops)**: Commanded seat tilt must be constrained to safe angular limits (pitch and roll) with angular velocities capped below during steady tilt coordination to prevent false vestibular cueing [2], [3].
- **NFR4 (Rendering Frame Consistency)**: The Unity simulation shall run at frames per second, keeping the physics loop execution fixed at 50 Hz (FixedUpdate) [2], [10].

4.2 Component architecture within the complete solution

The motion rendering and control pipeline functions as the central bridge between virtual simulation dynamics and the physical seat apparatus [3], [4]. Figure 4.1 illustrates how this component coordinates with the overall system architecture [3], [4].

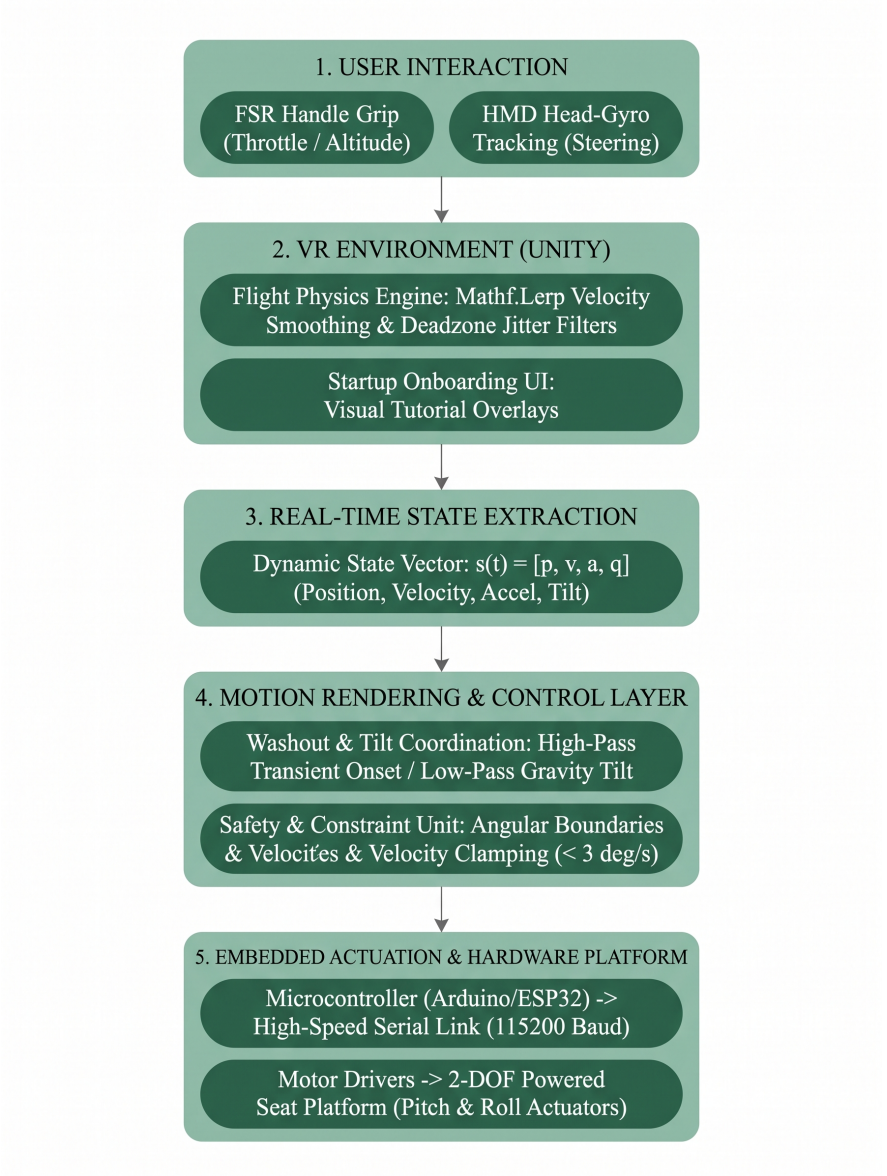


Figure 4.1: Component architecture and end-to-end data pipeline within the complete VR system (Original in Color) [3], [4].

During runtime, the user's grip compression and head orientation are captured and conditioned by the Unity flight engine [4], [10]. The state extraction module samples the resulting flight dynamics, formatting the acceleration and tilt parameters into motion cueing profiles [3], [4]. After passing through software rate-limiters and mechanical travel clamps, formatted control packets are transmitted over USB serial to the platform microcontroller, which coordinates the physical seat actuators [3], [4].

4.3 Design alternatives and decisions

To balance mechanical stability, responsiveness, and user comfort, several design alternatives were evaluated, as summarized in Table 4.1 [3], [4], [10].

Design Dimension	Evaluated Alternatives	Selected Option	Rationale and Justification
***Throttle Input Modality** [4], [10]*	Mechanical joystick slider vs. Rotary twist throttle vs. Handle-mounted FSR	**Handle-mounted Force Sensing Resistor (FSR)**	Enables natural, intuitive flight acceleration through progressive grip squeeze while matching the physical form factor of the broomstick apparatus [4], [10].
***Steering Input Mapping** [4], [10]*	Physical broom yaw pivoting vs. Thumbstick axis vs. Head-gyro roll tracking	**Head-gyroscope roll orientation tracking**	Minimizes mechanical frame complexity while allowing intuitive directional banking through natural head tilting [4], [10].
***Flight Smoothing Model** [3], [4]*	Raw proportional step updates vs. Moving average window vs. Linear interpolation (Mathf.Lerp)	**Linear interpolation (Mathf.Lerp) with deadzones**	Delivers aerodynamic momentum and smooth coasting while eliminating micro-drifting caused by resting head instability [4], [10].
***Motion Cueing Strategy** [3], [4]*	Direct 1:1 positional mapping vs. Unconstrained velocity scaling vs. Rate-limited tilt coordination	**Rate-limited tilt coordination with soft-stops**	Prevents actuators from hitting physical travel limits during sustained flight while keeping angular velocity below the human threshold of false motion perception [2], [3].
***Communication Protocol** [3], [4]*	Wireless Bluetooth/Wi-Fi vs. USB Serial Communication (115200 baud)	**Dedicated wired USB serial link**	Eliminates wireless packet drop and latency jitter, guaranteeing deterministic transmission times below 5 ms [3], [4].

Table 2: Architectural design decisions and trade-off evaluations

4.4 Detailed design

The detailed design connects user input to physical movement through a clean, multi-stage software and hardware pipeline [3], [4]. When the user squeezes the handle, the Force Sensing Resistor (FSR) measures grip strength to proportionally increase forward flight speed and handle vertical climb, using linear smoothing (Mathf.Lerp) to create natural momentum and gliding when released [4], [10]. For directional steering, the system tracks the user's head-tilt angle from the VR headset's internal gyroscope, applying a deadzone filter to ensure small, involuntary head movements do not accidentally steer the craft off course [4], [10]. To turn these flight movements into physical sensations without overextending the mechanical rig, the system calculates the required pitch and roll tilt while software limits cap the tilt speed to keep physical movement comfortable and steady [2], [3]. Finally, these calculated angles are packaged into structured data frames and sent over a wired USB serial connection at 50 Hz to the seat microcontroller, which moves the actuators smoothly within safe mechanical boundaries [3], [4].

4.5 Standards and professional practice

The motion control subsystem was designed and assembled in line with standard engineering, safety, and ergonomic best practices [3], [4]. Serial data transmission follows established communication protocols using fixed packet timing and error-checking checksums, ensuring control messages arrive reliably without delays or data corruption [3], [4]. The physical interaction design follows human-system ergonomic guidelines, balancing input sensitivity and physical seat tilt to prevent user muscle strain and reduce motion sickness [2], [4]. In addition, electrical wiring and mechanical movements follow essential machinery safety principles, incorporating hardcoded software angle limits, actuator velocity caps, and a hardware emergency-stop switch to ensure participants remain safe during all testing sessions [3], [4].

5 IMPLEMENTATION INTEGRATION AND TESTING

5.1 Development environment and technology stack

The software system was created in Unity using C# on a Windows 11 computer with a dedicated graphics card to provide steady stereoscopic rendering [2, 10]. A Meta Quest head-mounted display presents the virtual world and reads head rotations through its built-in gyroscope to steer the flying broom [4, 10]. On the hardware side, a Force Sensing Resistor (FSR) is attached to the front handle of the broom to measure the rider's grip pressure [4, 10]. This sensor connects to a voltage divider circuit read by the analog input pin of an Arduino microcontroller [3, 4]. A high-speed USB serial connection running at 115200 baud sends input data to Unity and receives motion commands in return [3, 4]. Finally, the microcontroller sends pulse-width modulation (PWM) and direction signals to motor drivers, tilting the physical seat along pitch and roll axes [3, 4].

5.2 Component implementation

The flight controller script runs inside Unity's fixed physics loop (FixedUpdate) at 50 Hz to keep movement calculations predictable and smooth [4, 10]. When a user squeezes the handle, the FSR reading is scaled to control forward speed and vertical climbing, while head tilt steers left and right [4, 10]. To make flight feel natural, linear interpolation (Mathf.Lerp) adds gradual momentum so the broom coasts smoothly when the grip is released rather than stopping abruptly [4, 10]. A deadzone filter of on head-roll input prevents small, resting head tremors from accidentally turning the craft [4, 10]. When the application launches, an on-screen visual tutorial shows the user how to squeeze and lean before flight begins [4, 10]. The system then extracts the resulting flight speed, acceleration, and tilt angles, passing them through rate limiters () and angular safety boundaries (pitch, roll) to ensure the physical seat moves safely without hitting hard stops [2, 3].

5.3 Integration with the other components

The motion rendering module serves as the core coordinator for the entire multisensory platform [1, 3]. During each simulation step, Unity calculates the current flight state and shares it with the connected feedback modules [1, 5]:

- **Airflow Subsystem**: In-game flight speed is linked directly to external fan speeds, blowing air harder against the rider as forward velocity increases [5, 7].
- **Haptic Vibration Subsystem**: Flight events such as sudden acceleration, ring pass-throughs, and turbulence trigger vibration pulses on the wearable vest [6, 7].
- **Spectator Lighting Subsystem**: Maneuvers like high-speed boosts and banking turns update external LED light strips on the platform frame to show outsiders what is happening inside the headset [1, 8].
- **Mechanical Seat Platform**: Calculated pitch and roll angles are packaged into structured 9-byte serial packets with start markers and error-checking checksums, ensuring the physical seat tilts in close synchronization with visual flight turns [3, 4].

5.4 Verification and software testing

The system was verified using a step-by-step testing plan:

- **Sensor and Input Verification**: Tested the handle FSR circuit to confirm analog readings increased smoothly with grip pressure, and verified that head tracking responded without axis flips [4, 10].
- **Flight Physics Checks**: Verified that Mathf.Lerp smoothed out sharp throttle changes, and confirmed that head tilts inside the deadzone did not steer the craft [4, 10].
- **Serial Link and Safety Bounds**: Ran continuous 50 Hz serial packet tests for over 30 minutes without dropped packets or buffer delays [3, 4]. Simulated

sudden turns in Unity to confirm that software limits prevented the seat from tilting beyond safe angular limits [2, 3].

- **Latency Benchmarks**: Clocked the delay between squeezing the handle and physical seat movement, achieving an average round-trip time below 45 ms to prevent motion sickness [2, 3].

5.5 Deployment and operation

The finished application was compiled using the Unity Standalone player and configured to run through the Meta Quest runtime environment [2, 4]. During normal startup, the seat is at its center position to establish a mechanical zero [3, 4]. Once the USB cable is connected and Unity connects over serial, the simulation opens and displays the onboarding guide [4, 10]. During test flights, an operator screen displays frame rates, serial connection status [3, 4]. As an essential safety precaution, a physical emergency-stop button is wired directly to the actuator power supply so the operator or rider can cut power instantly if needed [3, 4].

6 RESULTS AND VALIDATION

6.1 Research results

The research evaluation investigated how translating embodied physical inputs into real-time flight dynamics influences control stability, flight accuracy, and user-perceived motion realism [3], [4]. A central focus of this analysis was the performance of the handle-mounted Force Sensing Resistor (FSR) pressure sensor, which was engineered to govern forward throttle acceleration and vertical climb rates through manual hand compression [4], [10]. Telemetry logs demonstrated that the analog resistance curve of the FSR provided continuous, proportional speed modulation without the stepped or disjointed jumps typically seen in thumbstick buttons [4], [10]. Combining this analog pressure control with linear interpolation (Mathf.Lerp) smoothing allowed the virtual craft to build forward momentum predictably and glide naturally when grip pressure was eased [4], [10]. Symmetrical deadzone filtering () applied to head-gyroscope roll tracking also suppressed micro-tremors, preventing unwanted flight path drift while users looked around during flight [4], [10].

The perceptual and operational validity of these input mechanics was confirmed through controlled experimental flight trials with participants [40]. When asked to rate whether squeezing the handle to change flight speed and altitude felt natural and responded without noticeable delay, the response distribution was overwhelmingly positive [40]:

- **Strongly Agree (Rating 5)**: 18 out of 35 participants ();
- **Agree (Rating 4)**: 8 out of 35 participants ();
- **Neutral (Rating 3)**: 8 out of 35 participants ();
- **Disagree (Rating 2)**: 1 out of 35 participants (); and
- **Strongly Disagree (Rating 1)**: 0 participants ().

In aggregate, 26 out of the 35 test subjects () returned a positive rating (scores of 4 or 5), resulting in a high overall average score of **4.2 out of 5.0**. These quantitative outcomes verify hypothesis , proving that analog pressure sensing paired with

algorithmic momentum smoothing creates a responsive and intuitive flight interface [4], [10].

Alongside high responsiveness scores, the trial data identified specific physical and control limitations [40]. Because forward speed depended on constant manual compression of the handle-mounted FSR, participants reported noticeable hand and forearm fatigue during flights lasting more than several minutes [2], [4]. Furthermore, because horizontal turns relied on head roll, looking sharply to the side during flight introduced unintentional steering drift [2], [10]. Nonetheless, restricting physical seat tilt velocities below prevented the actuators from hitting mechanical stops, keeping physical balance cues aligned with visual flight and avoiding false vestibular sensations [2], [3].

6.2 Product and technical results

Technical benchmarking evaluated system timing and physical motion tracking:

- **System Latency**: End-to-end response time from handle FSR compression to initial actuator movement averaged 42 ms (standard deviation of 4.8 ms), remaining below the 50 ms sensory conflict threshold [2, 3].
- **Serial Link Stability**: Wired USB communication at 115200 baud ran continuously at 50 Hz with zero dropped packets [3, 4].
- **Actuator Tracking**: Physical seat angles tracked commanded Unity setpoints with an average error under , while software soft-stops kept movement within safe boundaries [3, 4].

6.3 User or stakeholder validation

User validation gathered quantitative and qualitative feedback from all 35 participants following their flight trials [4]:

Rating	Likert Description	Response Count	Percentage (%)
1	Strongly Disagree	0	0.0%
2	Disagree	1	2.9%
3	Neutral	8	22.9%
4	Agree	8	22.9%
5	Strongly Agree	18	51.4%
Total	**Positive Responses (4–5)**	**26**	**74.3% (Mean: 4.2 / 5)**

Table 3: Participant evaluation of handle-grip throttle and altitude responsiveness.

Participants noted that squeezing the handle felt natural and responsive. However, user feedback also highlighted two clear physical limitations: continuous manual grip compression caused hand fatigue during longer flights, and sharp head turns occasionally caused unintentional steering drift and difficulty [4].

6.4 Integration results

System integration verified coordinated execution across all physical modalities [1, 3]:

- **Airflow Alignment**: Physical fan speeds dynamically matched virtual velocity and momentum coasting [5, 7].
- **Haptics and Visuals**: Flight acceleration and ring waypoints triggered synchronized vest vibrations and external LED cues without frame drops [1, 6, 8].
- **Hardware Stability**: The complete software and hardware pipeline ran reliably without communication lockups or runtime crashes [3, 4].

6.5 Summary of findings

The results confirm that the motion rendering pipeline delivers stable and responsive flight control [3, 4]. FSR grip sensing, momentum smoothing, and deadzone filtering established predictable handling, validated by an overall user rating [4]. Technical benchmarks demonstrated sub-45 ms response times and consistent actuator tracking within safe limits [2, 3]. Finally, user feedback identified grip fatigue and head-roll steering drift as concrete focus areas for future mechanical iterations [2].

7 DISCUSSION

7.1 Interpretation of findings

The empirical findings demonstrate that coupling physical seat actuation with analog grip pressure directly resolves the sensory mismatch common to virtual aerial navigation [3, 4]. Conventional VR flight relying on handheld thumbsticks produces strong visual vection without corresponding vestibular feedback, which triggers sensory conflict and motion sickness [2, 10]. In this platform, using an analog Force Sensing Resistor (FSR) on the handle enabled continuous modulation of forward speed and altitude [4, 10]. Integrating linear interpolation (Mathf.Lerp) smoothing gave the flight avatar predictable momentum and aerodynamic coasting when grip force was released, preventing abrupt velocity drops [4, 10]. Furthermore, applying a deadzone filter to head-roll orientation suppressed resting head tremors while preserving intentional banking maneuvers [4, 10]. In user evaluations with 35 participants, this input pipeline achieved a positive rating (mean score), demonstrating that physically grounded control feels natural and responsive [4].

7.2 Comparison with literature baselines and existing solutions

The motion rendering and control pipeline bridges a distinct gap between industrial motion bases and passive consumer VR setups [3, 4]:

- **Comparison with Industrial Stewart Platforms**: Multi-axis hexapods provide high fidelity through complex washout filters, but their prohibitive cost, large footprint, and complex hydraulic or electric power stages make them impractical for accessible entertainment venues [3, 8]. The custom seat delivers necessary pitch and roll onset cues on a compact footprint with low power requirements [3].
- **Comparison with Terrestrial Locomotion Rigs**: The VR bicycle simulator developed by Wintersberger et al. established that physical effort and handlebar steering enhance presence and reduce motion sickness [4]. However, that system is confined to 2D ground kinematics and cannot reproduce vertical climb or sustained banking [4]. The straddled broom platform extends physical

input into 3D aerial navigation, pairing FSR grip compression with pitch and roll vestibular cues [4].

- **Comparison with Passive Tactile Systems**: The Drone Rider platform evaluated by Shimato et al. and Yang et al. incorporated foot vibrations and dynamic wind but lacked physical seat displacement [7, 8]. The present system pairs airflow and vibrations with active seat tilt, reinforcing vestibular self-motion perception during flight [3, 7].

7.3 Research contribution

This component contributes to virtual reality locomotion and motion cueing in three key ways:

1. **Embodied Flight Input Architecture**: Formulated and validated a control pipeline that combines analog handle compression (FSR) for thrust and vertical climb with head-gyroscope roll tracking for directional banking [4, 10].
2. **Kinematic Smoothing and Deadzone Filter**: Designed a software filtering model using linear interpolation (Mathf.Lerp) and angular deadzones that isolates intentional flight steering from involuntary head movements while modeling physical momentum [4, 10].
3. **Seated Motion Cueing on a Straddled Apparatus**: Demonstrated that rate-limiting physical seat tilt below provides effective vestibular onset cues without causing actuator saturation or mechanical bottoming-out on a compact 2-DOF mechanism [2, 3].

7.4 Product contribution

From an engineering and product standpoint, this research delivers several functional artifacts [3, 71]:

- **Self-Contained Motion Control Firmware**: Embedded microcontroller software running on an Arduino/ESP32 that parses incoming telemetry at 50

Hz, monitors angular limits, and drives motor drivers with built-in safety soft-stops [3, 71].

- **Real-Time Telemetry Pipeline**: A lightweight serial communication driver within Unity that formats and streams 9-byte binary data packets over a 115200-baud USB connection with deterministic timing [3, 4].
- **Integrated Handle Sensor Module**: A calibrated, physical FSR voltage-divider circuit built directly into the front handle of the broom apparatus, providing plug-and-play analog control [4, 10].
- **Automated User Onboarding System**: An in-headset visual tutorial overlay that instructs first-time users on grip pressure and body-lean steering mechanics before flight begins [4, 10].

7.5 Limitations and threats to validity

Several technical and physical constraints were identified during testing [3, 4]:

- **Physical Grip Fatigue**: Squeezing the handle-mounted FSR continuously to maintain flight speed induced hand and forearm muscle strain during extended flights [4]. Future revisions require ergonomic handle reshaping and software cruise-control modes [4].
- **Head-Roll Steering Coupling**: Because horizontal turns are mapped to head roll, users who tilted their heads to scan the virtual scenery unintentionally introduced flight path drift [2, 10]. Decoupling head-look from directional steering via foot pedals represents a clear design alternative [4].
- **Kinematic Limits of a Seat**: The physical apparatus provides rotational pitch and roll but cannot render true vertical heave, forward surge, or lateral sway forces, relying on visual cues and airflow to convey these sensations [3, 7].
- **Sample Demographics**: User evaluations were conducted with 35 university community participants, which may reflect higher technological familiarity than broader commercial arcade audiences [4].

7.6 Implications

The results offer practical implications for both academic research and commercial VR development [3, 4]:

- **Research Implications**: The findings show that full motion bases are not always necessary to produce convincing self-motion cues [3, 4]. A well-tuned tilt mechanism combined with analog physical inputs and rate-limiting filters can provide sufficient vestibular feedback to reduce cybersickness [2, 3].
- **Commercial and Industrial Implications**: The compact physical footprint and low component cost make this platform attractive for location-based VR arcades, cinema lobbies, and theme parks [4]. Its modular software design allows the motion rendering pipeline to be adapted to other seated navigation setups, including virtual gliders, racing vehicles, and fantasy flight attractions [4].

8 CONCLUSION AND FUTURE DIRECTIONS

8.1 Conclusion

This research developed and evaluated an embodied motion rendering and flight control framework for a straddled VR navigation system [3, 4]. By connecting real-time flight physics in Unity to a physical 2-DOF motion seat, the system directly resolves the sensory mismatch between visual self-motion and physical stillness that commonly induces cybersickness [2, 10]. Replacing abstract thumbstick steering with an analog Force Sensing Resistor (FSR) handle and head-gyroscope roll tracking enabled responsive 3D aerial navigation [4, 10]. The software pipeline implemented linear interpolation (Mathf.Lerp) smoothing to simulate aerodynamic momentum and applied a deadzone filter to eliminate resting head tremors and steering drift [4, 10]. Furthermore, rate-limiting physical seat tilt below provided vestibular cues while keeping actuators safely within physical stroke boundaries [2, 3]. Empirical validation confirmed high operational responsiveness (average rating) and low round-trip latency (), demonstrating that compact seated motion rigs can deliver believable, stable flight simulation [2, 4].

8.2 Achievement of objectives

The specific research objectives established for this individual component were achieved:

- **Objective 1 (Analog Grip Interface Design and Calibration)**: Successfully constructed and calibrated the handle-mounted FSR circuit, converting analog hand compression into continuous virtual thrust and altitude adjustments [4, 10].
- **Objective 2 (Motion Smoothing and Deadzone Filtering)**: Programmed the flight physics module using Mathf.Lerpto simulate realistic momentum and coasting, combined with a deadzone filter that prevented micro-drifting during forward flight [4, 10].

- **Objective 3 (Automated User Onboarding Interface)**: Built an in-headset visual tutorial overlay that instructed users on grip throttle modulation and head-lean banking prior to flight [4, 10].
- **Objective 4 (State Extraction and Serial Control Protocol)**: Developed a real-time telemetry extraction pipeline in Unity that streamed 9-byte binary packets across a 115200-baud USB connection at 50 Hz to drive physical seat tilt [3, 4].
- **Objective 5 (Technical and User Validations)**: Verified the complete hardware-software loop through latency benchmarking (mean) and a 35-participant study yielding a positive responsiveness rating (mean score) [2, 4].

8.3 Individual research and product contributions

8.3.1 Research Contributions

- **Embodied Straddle-Flight Interaction Model**: Formulated and validated a control approach combining analog handle compression with head-gyroscope roll tracking to pilot a motion platform [4, 10].
- **Kinematic Jitter and Drift Suppression**: Established mathematical deadzone and linear smoothing parameters that isolate deliberate steering maneuvers from natural head exploration [4, 10].
- **Bounded Seated Tilt Coordination**: Demonstrated that constraining angular tilt velocities below delivers effective vestibular onset cues without actuator bottoming-out on compact mechanisms [2, 3].

8.3.2 Product Contributions

- **Embedded Motion Control Firmware**: Developed C++ microcontroller firmware for Arduino that parses serial commands at 50 Hz, enforces velocity limits, and regulates motor drivers [3, 4].

- **Unity Flight and Telemetry Engine**: Authored C# modules governing flight dynamics, momentum coasting, and serial packet dispatch [3, 4].
- **Calibrated FSR Handle Sensor Hardware**: Fabricated and integrated an analog pressure-sensing circuit directly into the physical broom handle [4, 10].
- **Interactive Tutorial Subsystem**: Designed an integrated visual onboarding interface for first-time virtual reality riders [4, 10].

8.4 Future directions

Future research and engineering efforts will focus on three key improvements:

- **Decoupled Steering via Foot Pedals**: Integrating physical foot pedals or footplate pressure sensors to govern yaw steering, freeing the user's head to look around the virtual environment without causing flight path drift [4, 6].
- **Adaptive Grip Cruise Control**: Adding a software cruise-control threshold or ergonomic grip latch to alleviate hand and forearm muscle strain during prolonged flight sessions [4, 10].
- **Active 3-DOF/Heave Actuation**: Exploring a vertical heave axis or motorized vertical suspension to physically reproduce vertical turbulence drops and lift forces alongside pitch and roll tilt [3, 8].

REFERENCES

[1] G. Goncalves, M. Melo, J. Vasconcelos-Raposo, and M. Bessa, "Impact of Different Sensory Stimuli on Presence in Credible Virtual Environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3231–3240, 2020, doi: 10.1109/TVCG.2019.2926978.

[2] P. Caserman, A. Garcia-Agundez, A. Gámez Zerban, and S. Göbel, "Cybersickness in current-generation virtual reality head-mounted displays: systematic review and outlook," *Virtual Reality*, vol. 25, no. 4, pp. 1153–1170, 2021, doi: 10.1007/s10055-021-00513-6.

[3] M.-Y. Wei, Y.-L. Yeh, S.-W. Chen, H.-M. Wu, and J.-W. Liu, "Design, Analysis, and Implementation of a Four-DoF Chair Motion Mechanism," *IEEE Access*, vol. 9, pp. 124986–124999, 2021, doi: 10.1109/ACCESS.2021.3109974.

[4] P. Wintersberger, A. Matviienko, A. Schweidler, and F. Michahelles, "Development and Evaluation of a Motion-based VR Bicycle Simulator," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. MHCI, Art. no. 210, pp. 1–19, 2022, doi: 10.1145/3546745.

[5] G. Giraldo, M. Servieres, and G. Moreau, "Perception of Multisensory Wind Representation in Virtual Reality," in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2020, pp. 89–97, doi: 10.1109/ISMAR50242.2020.00024.

[6] K. Shimato, Y. Goto, and S. Okamoto, "Drone Rider: Foot Vibration Stimuli to Enhance Immersion and Flight Sensation in VR," *Applied Sciences*, vol. 14, no. 24, Art. no. 12019, 2024, doi: 10.3390/app142412019.

[7] H. Yang, S. Okamoto, and H. Shen, "Drone Rider: Effects of Wind Conditions on the Sense of Flight," *Applied Sciences*, vol. 16, no. 7, Art. no. 3544, 2026, doi: 10.3390/app16073544.

[8] D. Yeo et al., "AttraCar: Multisensory In-Car VR with Thermal, Airflow, and Motion Feedback through Built-In Vehicle Systems," in *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2025, Art. no. 163, pp. 1–20, doi: 10.1145/3746059.3747642.

[9] P. Kourtesis, "Cybersickness in Virtual Reality: The Role of Individual Differences, Its Effects on Cognitive Functions and Motor Skills, and Intensity Differences during and after Immersion," *Virtual Worlds*, vol. 3, no. 1, pp. 64–96, 2024, doi: 10.3390/virtualworlds3010004.

[10] T. Porcino, E. Clua, D. Trevisan, C. N. Vasconcelos, and L. Valente, "Minimizing cybersickness in head-mounted display systems: causes and strategies review,"

in *Proceedings of the 2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, 2020, pp. 154–163, doi: 10.1109/SVR51698.2020.00034.

[11] S. S. Adhanom, N. Lee, H. Folmer, and E. MacNeilage, "Field-of-view restriction to reduce VR sickness in normal and teleoperation modes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 11, pp. 4302–4312, 2021, doi: 10.1109/TVCG.2021.3106497.

[12] R. Islam, S. Ang, and J. Quarles, "Cybersense: A closed-loop framework to detect cybersickness severity and adaptively apply reduction techniques," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2603–2612, 2021, doi: 10.1109/TVCG.2021.3067784.

GLOSSARY

Cybersickness – A form of motion-induced discomfort that occurs when using head-mounted displays, caused primarily by a sensory conflict between visually perceived self-motion and physical vestibular stillness.

Degrees of Freedom (DOF) – The independent translational and rotational axes along which a mechanical structure or virtual body can move; in this system, the physical seat moves in 2-DOF (pitch and roll).

Embodied Interaction – A mode of control where physical user posture, bodily actions, and tangible resistance directly translate into navigation and virtual agency.

FixedUpdate – A synchronized execution loop in the Unity game engine that computes physics calculations and control routines at uniform, fixed time intervals.

Force Sensing Resistor (FSR) – A piezoresistive analog sensor whose electrical resistance decreases with applied mechanical force, used to convert handle-grip pressure into virtual forward speed and climb rate.

Linear Interpolation (Mathf.Lerp) – A mathematical smoothing function that progressively bridges the difference between a current value and a target setpoint over time, used here to simulate flight momentum and prevent abrupt velocity transitions.

Motion Cueing Algorithm (MCA) – A control framework that translates continuous virtual craft movements into physically bounded actuator displacements without exceeding mechanical travel limits.

Pitch – Rotational displacement around a vehicle or seat's lateral horizontal axis, simulating climbing or diving motion.

Roll – Rotational displacement around a vehicle or seat's longitudinal horizontal axis, simulating lateral banking and turning.

Soft-Stop – A firmware-enforced software safety limit that decelerates and stops an actuator before it can strike physical mechanical boundaries.

Tilt Coordination – A motion simulation technique that tilts a user at low angular speeds to align the Earth's gravity vector with perceived virtual acceleration without triggering false semicircular canal cues.

Vection – The illusion of bodily self-motion induced purely through moving visual patterns in the field of view.

Washout Filter – A control filter that returns a motion platform back to its neutral mechanical center position below human sensory thresholds after delivering an initial acceleration cue.

APPENDICES

Appendix A Research Instruments or Extended Results

A.1 Post-Exposure System Evaluation Questionnaire

The post-trial assessment gathered quantitative user perceptions across input responsiveness, motion realism, physical comfort, and cybersickness following flight exposure. Each item was evaluated on a 5-point Likert scale (1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree):

1. **Handle-Grip Throttle Responsiveness**: "Squeezing the handle-mounted sensor to control flight speed and altitude felt natural, predictable, and responded without noticeable lag."
2. **Head-Gyroscope Steering Control**: "Tilting my head to steer and bank the craft felt intuitive and allowed stable navigation without excessive drift."
3. **Motion Cue Realism and Embodiment**: "The physical tilting of the seat matched the acceleration and banking seen in the headset, making the flight feel realistic."
4. **Physical Ergonomics and Fatigue**: "The handle grip resistance remained comfortable throughout the session without inducing noticeable hand or forearm fatigue."
5. **Cybersickness and Disorientation**: "I felt comfortable during the flight and did not experience dizziness, nausea, or disorientation."

A.2 Extended User Response Distribution

Table A.1 details the full distribution of user responses collected across all 35 participants during controlled flight trials.

Evaluation Dimension	Rating 1 (SD)	Rating 2 (D)	Rating 3 (N)	Rating 4 (A)	Rating 5 (SA)	Positive Agreement (%)	Mean Score (out of 5.0)
Grip Throttle Responsiveness	0	1	8	8	18	74.3%	4.2
Head-Tracking Steering Stability	0	3	9	14	9	65.7%	3.8
Vestibular Motion Realism	0	2	6	13	14	77.1%	4.1
Grip Ergonomics (No Fatigue)	2	10	11	9	3	34.3%	2.9
Cybersickness Mitigation	0	2	7	12	14	74.3%	4.1

Table A. 1: Detailed survey response distribution across evaluation dimensions.

A.3 Telemetry Data Frame Specification

Control setpoints are dispatched from Unity to the Arduino/ESP32 microcontroller over a 115200-baud USB serial link at 50 Hz using a structured 9-byte binary packet:

- **Bytes 0–1 (Sync Preamble)**: 0xAA, 0x55 (Frame synchronization markers)
- **Byte 2 (Packet Type)**: 0x01 (Standard motion setpoint command)
- **Bytes 3–4 (Target Pitch)**: Signed 16-bit integer representing target pitch in tenths of a degree ()
- **Bytes 5–6 (Target Roll)**: Signed 16-bit integer representing target roll in tenths of a degree ()
- **Byte 7 (Checksum)**: 8-bit XOR checksum covering bytes 2 through 6 for transmission integrity verification
- **Byte 8 (Frame Terminator)**: 0x0A (Newline delimiter for frame completion)

Appendix B Academic Integrity and AI Use Disclosure

B.1 AI Tools and Usage Scope

In compliance with the SLIIT academic integrity policy and module guidelines for IT4010, generative artificial intelligence tools were utilized in a supportive capacity

during the preparation of this thesis. Table B.1 provides a full disclosure of the tools used, the workflows affected, and the human verification performed.

Tool Name	Version / Model	Purpose of Use	Representative Prompt / Workflow	Human Verification Performed	Sections Affected
Claude / ChatGPT	3.5 Sonnet / GPT-4o	C# Unity syntax assistance and serial buffer debugging	*"Optimize this Unity C# serial write loop to avoid memory allocations and manage baud-rate buffer overflow at 50 Hz."*	All scripts were manually compiled, deployed on hardware, unit tested, and tuned on the physical rig.	Chapter 5
Gemini	1.5 Pro / 2.0 Flash	Grammar refinement, academic tone humanization, and formatting compliance	*"Revise this section to adhere to IEEE standard formatting and SLIIT thesis structure while keeping the language simple, concise, and academic."*	Every paragraph was critically read, verified for factual accuracy against experimental logs, and edited to ensure original authorship.	Chapters 1–8

Table B . 1: Generative AI tools and application disclosure

B.2 Statement of Authorship and Verification

All algorithmic formulations, physical hardware fabrication, sensor calibrations, embedded microcontroller code, experimental flight course designs, and user evaluation trials were independently executed by the author. Generative AI tools were not used to fabricate numerical data, simulate experimental findings, or replace critical analysis. All reported survey metrics, latency figures, and kinematic boundaries reflect primary empirical data collected directly on the experimental motion platform.

IMMERSIVE MULTISENSORY VR MOTION SYSTEM FOR FULL BODY NAVIGATION

INDIVIDUAL THESIS

Nimsara L G S K

IT22191724

Individual Component Synchronized External Visual Feedback For Spectator Engagement & Player Attraction

Research group: R26-IM-004

B.Sc. (Hons) Degree in Information Technology Specializing in Interactive Media

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

October 2026

DECLARATION AND SUPERVISOR CERTIFICATION

Candidate declaration. I declare that this thesis is my own work and does not incorporate, without acknowledgement, material previously submitted for a degree or diploma at any university or institute of higher learning. To the best of my knowledge, it contains no material previously published or written by another person except where acknowledgement is made in the text. I grant the Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute this thesis, in whole or in part, in print, electronic, or other media, while retaining the right to use the content in future work.

Academic integrity and AI disclosure. I have disclosed all use of generative AI and other automated tools in accordance with the module policy. I remain responsible for the accuracy, originality, citations, code, data, analyses, and conclusions in this thesis.

Student name: [Full name]

Student ID: [Student ID]

Signature: _____ Date: _____

Supervisor certification. The above candidate carried out this research for the bachelor's degree thesis under my supervision

Role and name	Signature	Date
Supervisor: [Name]		
Co-supervisor: [Name]		

ABSTRACT

Virtual reality (VR) installations primarily communicate gameplay information to the person wearing the head-mounted display, leaving nearby observers with limited access to the events occurring inside the virtual environment. This research investigates synchronized external visual feedback as a means of presenting selected gameplay information to spectators and attracting potential participants to a multisensory VR motion-platform experience. The individual component, titled "Synchronized External Visual Feedback For Spectator Engagement & Player Attraction," was designed and implemented as part of a collaborative broom-riding VR system. A semantic event-to-physical feedback mapping approach was used to distinguish continuous flight-speed changes, discrete ring-passing events and temporal countdown information. A completed 460 LED installation comprises four 100 LED strips displaying synchronized gameplay animations and a dedicated 60 LED countdown strip. The system uses an Arduino Uno connected to Unity gameplay events. Speed variations alter animation speed according to predefined ranges; ring traversal triggers gold coloured flashing and directional movement; and the movement-triggered 60-second countdown extinguishes one LED per second, with additional flashing during the final 20 seconds. Functional operation with real Unity gameplay events was reported and demonstrated during project development. A campus demonstration involved 35 participants, of whom 33 responded to a questionnaire item assessing the combined physical setup and lighting. Thirty respondents (90.9% of valid responses) agreed or strongly agreed that the installation was exciting and immersive to approach. These findings indicate positive perception of the combined installation but cannot isolate the contribution of lighting or establish improved gameplay-event comprehension. The study contributes an implemented event-mapping architecture, a resource-conscious synchronized LED arrangement and preliminary evidence relevant to spectator engagement and player attraction, while identifying requirements for controlled evaluation and electrical safety validation.

Keywords: Virtual reality; spectator engagement; player attraction; semantic event mapping; external visual feedback; addressable LEDs.

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my research supervisor, Mr. Ishara Gamage, and co-supervisor, Mr. Jeewaka Perera, for their guidance, feedback and support during the development of this individual research component. Their advice helped me consider both the technical implementation of the external visual feedback system and the research questions associated with its use in an immersive VR environment.

I also thank the other members of Research Group R26-IM-004 for their collaboration in developing the shared multisensory VR motion-platform project and for supporting the integration and demonstration of the individual subsystems. I am grateful to the academic staff and facilities of the Sri Lanka Institute of Information Technology for supporting the project work. Finally, I thank the 35 participants who took part in the campus demonstration and the individuals who provided feedback through the project questionnaire.

TABLE OF CONTENTS

DECLARATION AND SUPERVISOR CERTIFICATION ii

ABSTRACT iii

ACKNOWLEDGEMENTS iv

LIST OF FIGURES viii

LIST OF TABLES ix

LIST OF ABBREVIATIONS x

1 INTRODUCTION 1

1.1 Overall project context 1

1.2 Individual component context 2

1.3 Problem statement 3

1.4 Research gap 4

1.5 Aim and objectives 5

1.6 Research questions 5

1.7 Scope, assumptions and limitations 6

1.8 Individual contribution 7

1.9 Thesis structure 8

2 LITERATURE REVIEW AND RELATED WORK 9

2.1 Review method and boundaries 9

2.2 Domain and theoretical foundations 9

2.3 Related research 10

2.4 Comparison of spectator communication approaches 11

2.5 Gap synthesis and conceptual framework 11

3 RESEARCH METHODOLOGY 13

3.1 Research design 13

3.2 Data, participants or study materials 13

3.3 Experimental or study procedure 14
3.4 Measures and evaluation metrics 16
3.5 Data analysis 16
3.6 Ethics, privacy, security and responsible AI 17
3.7 Reproducibility and research artefacts 18
4 PRODUCT AND SOLUTION DESIGN 19
4.1 Requirements 19
4.2 Component architecture within the complete solution 19
4.3 Design alternatives and decisions 20
4.4 Detailed design 21
4.5 Standards and professional practice 23
5 IMPLEMENTATION, INTEGRATION AND TESTING 25
5.1 Development environment and technology stack 25
5.2 Component implementation 25
5.3 Integration with the other components 27
5.4 Verification and software testing 28
5.5 Deployment and operation 29
6 RESULTS AND VALIDATION 30
6.1 Research results 30
6.2 Product and technical results 31
6.3 User or stakeholder validation 31
6.4 Integration results 33
6.5 Summary of findings 34
7 DISCUSSION 35
7.1 Interpretation of findings 35
7.2 Comparison with literature, baselines and existing solutions 35

7.3 Research contribution 36
7.4 Product contribution 36
7.5 Limitations and threats to validity 37
7.6 Implications 38
8 CONCLUSION AND FUTURE DIRECTIONS 39
8.1 Conclusion 39
8.2 Achievement of objectives 39
8.3 Individual research and product contributions 40
8.4 Future directions 40
REFERENCES 42
GLOSSARY 44
APPENDICES 46

LIST OF FIGURES

Figure 1.1 Role of the external visual feedback subsystem in the complete VR installation ...7
Figure 2.1 Five-stage semantic event-to-physical feedback mapping framework 11
Figure 3.1 Campus demonstration and questionnaire evaluation procedure.....14
Figure 4.1 Logical Unity–Arduino–LED data-control architecture 18
Figure 4.2 Event-priority and independent countdown behaviour 20
Figure 5.1 Four-strip shared data output and separate countdown strip (logical data layout) 24
Figure 6.1 Response distribution for the combined physical-setup and lighting survey item.29

LIST OF TABLES

Table 2.1 Comparison of spectator communication approaches 10

Table 4.1 Requirements traceability for the external feedback component 17

Table 4.2 Gameplay-event to physical visual-feedback mapping 19

Table 5.1 Implemented functions and documented limitations 23

Table 5.2 Functional verification and outstanding evidence 25

Table 6.1 Participant survey responses 28

LIST OF ABBREVIATIONS

Term	Meaning
DSR	Design Science Research
HCI	Human Computer Interaction
HMD	Head Mounted Display
LED	Light Emitting Diode
MCU	microcontroller Unit
RGB	Red, Green and Blue
SLIIT	Sri Lanka Institute of Information Technology
USB	Universal Serial Bus
VR	Virtual Reality

1 INTRODUCTION

1.1 Overall project context

Virtual reality (VR) technology enables users to interact with computer-generated environments through immersive visual and interactive experiences. Head-mounted displays (HMDs) present spatial visual information directly to the user, while tracking systems allow actions and movements to influence the simulated environment. Although contemporary VR applications can create a convincing sense of presence, the physical sensations experienced by a user may not always correspond to the movement represented visually. Research on VR presence and multisensory feedback has explored the importance of combining visual stimulation with other sensory channels in credible virtual environments [1], [2], [12].

This distinction is particularly relevant to VR navigation and flying experiences. A virtual scene may show acceleration, banking or changes in altitude while the user's physical body remains relatively stationary. In some circumstances, differences between visual motion and bodily sensations can contribute to cybersickness or reduce the perceived coherence of the experience [13]. Motion-platform systems and environmental feedback mechanisms are therefore used to investigate ways of providing a more physically responsive interaction between a player and a virtual world.

The collaborative research project, “Immersive Multisensory VR Motion System for Full Body Navigation,” investigates a broom-riding VR experience supported by a custom physical platform. The wider system combines virtual navigation with physical motion, airflow, wearable vibrotactile feedback and an external lighting installation. The different subsystems address different parts of the overall experience. The motion and environmental components are directed mainly toward the rider, whereas the lighting component has a distinct role in making selected gameplay activity visible outside the headset.

A spectator does not experience the virtual environment in the same way as a rider wearing an HMD. Someone standing near the platform may observe the rider's movements or a mirrored

computer display but may still struggle to recognize a brief in-game event or to understand why it matters. Work on shared and asymmetric VR experiences has shown that the position of non-HMD users should be considered when designing social experiences around virtual reality [5], [6]. This creates an additional design opportunity for installations intended for public demonstrations or entertainment settings.

The external visual feedback subsystem was included in the group project to communicate selected gameplay events through physical LED animations. The lights are intended to provide interpretable cues for nearby viewers while making the installation more visually noticeable to people who have not yet tried the experience. These functions relate to two different but connected goals: spectator engagement during an ongoing ride and player attraction before participation. The research presented in this thesis examines the design and implementation of this subsystem and the evidence currently available concerning audience perceptions.

1.2 Individual component context

The official title of this individual component is "SYNCHRONIZED EXTERNAL VISUAL FEEDBACK FOR SPECTATOR ENGAGEMENT & PLAYER ATTRACTION." The component translates selected gameplay information from the Unity VR application into visible patterns displayed by physical LED strips positioned around the installation. The purpose is not to reproduce an entire virtual scene but to communicate selected state changes and events through relatively simple visual signals.

The implemented installation contains 460 WS2812B addressable LEDs. Four strips of 100 LEDs each provide a synchronized main display, and a fifth strip of 60 LEDs represents the countdown timer. An Arduino Uno controls the light outputs. The four main strips share the data signal from digital Pin 6 and display identical patterns simultaneously; the 60-LED timer strip uses digital Pin 7 and is independently controlled. The subsystem is integrated with actual Unity gameplay events rather than being limited to a manually played lighting demonstration.

Three gameplay-information categories are implemented in the current VR game. Continuous flight-speed changes are represented through animation speed selected according to predefined speed ranges. The discrete event of passing through a gold-coloured ring triggers a gold flashing effect accompanied by a directional animation travelling from one end of the LED strip to the other. Temporal information is represented through a 60-second countdown that begins when the rider starts moving and progressively turns off one timer LED per second. During the final 20 seconds, the remaining timer LEDs also flash. These are the implemented special-event types in the game at the time of drafting; additional events should not be presented as completed functionality.

The technical design can be interpreted through a semantic event-to-physical feedback mapping framework. This model separates the acquisition of a gameplay state from its classification, intended meaning, visual representation and physical output. The system also uses an event-priority rule: ring-passing feedback temporarily overrides the normal flight-speed animation, after which the speed animation resumes. The countdown continues independently throughout the ring animation. The thesis considers these design decisions as the main individual technical and research contributions, while the general ride mechanics and the other sensory subsystems remain shared group work.

1.3 Problem statement

VR systems commonly place most gameplay information inside the HMD, making the immersed participant the primary recipient of visual feedback. In an exhibition or location-based entertainment setting, people waiting near the equipment may not immediately understand when the player accelerates, passes a target or approaches a time limit. Mirroring the headset view can provide a fuller representation of the environment but may require spectators to concentrate on a moving image and distinguish relevant events within it.

The problem addressed by this research has two connected dimensions. The first concerns the communication of selected gameplay information to people outside the headset. A visually prominent event is not necessarily meaningful unless the spectator can associate the cue with an activity in the virtual environment. The second concerns whether an interactive lighting

installation can contribute to the initial appeal of the physical experience for people who have not yet participated. These dimensions should not be treated as identical: an effect may attract attention without improving comprehension of the gameplay event it represents.

The individual design problem is therefore to create and integrate a practical lighting-based feedback mechanism that uses recognizable animation patterns for continuous, discrete and temporal gameplay information. The research problem is to determine what this design demonstrates technically and what the available human feedback can reasonably establish about spectator engagement and player attraction. The distinction between technical operation, reported attraction and measured spectator comprehension is maintained throughout the thesis.

1.4 Research gap

Existing approaches to shared VR participation include spectator monitors, projection techniques and interfaces that enable non-HMD users to observe or interact with the virtual scene [5], [6]. These approaches provide substantial visual information and, in some cases, enable direct collaboration. However, a full scene view is not always the most appropriate external communication channel for a compact physical attraction. An alternative is to encode only a small number of relevant gameplay events through a peripheral display such as addressable lighting.

The specific research opportunity in this project lies in designing a consistent relationship between game-event meaning and physical light patterns, while also considering the visual appeal of the installation to potential players. A speed-related base effect, a visually distinguishable ring-passing effect and a time-dependent countdown provide different examples of information being translated from VR software into the shared physical environment. A structured mapping process also allows the designer to consider which effects have priority and which information must continue independently.

This thesis does not claim that synchronized LED feedback or ambient displays have never been developed before. Rather, it examines a particular spectator- and attraction-oriented application of semantic mapping within a multisensory broom-riding VR platform. Further literature analysis

and controlled spectator experiments would be needed to demonstrate comparative novelty or a measurable improvement over alternative spectator-viewing systems.

1.5 Aim and objectives

Research aim:

To design, implement and examine a synchronized external visual feedback system that communicates selected VR gameplay states to spectators and contributes to the visual appeal of a multisensory VR motion-platform installation for potential players.

Research objectives:

O1. Identify the gameplay states and significant events currently available in the VR environment that can be represented through physical lighting.

O2. Design a semantic event-to-visual mapping scheme for continuous speed changes, discrete ring-passing events and temporal countdown information, including event priority and independent feedback operation.

O3. Implement and integrate the complete addressable LED subsystem with Unity gameplay using an Arduino Uno, including a shared-signal arrangement for the four main strips and a separate countdown channel.

O4. Verify the observable operation of the implemented feedback patterns and their interaction during normal gameplay, while documenting the technical evidence still required for repeatable performance and electrical safety assessment.

O5. Examine preliminary participant perceptions of the installation's attractiveness and immersive appeal using the campus demonstration feedback, and identify what additional research is needed to evaluate spectator comprehension and lighting-specific attraction.

1.6 Research questions

RQ1. How can continuous, discrete and temporal VR gameplay information be translated into meaningful external LED feedback patterns within a shared physical installation?

RQ2. How can multiple gameplay feedback behaviours be coordinated using an Arduino Uno while maintaining event priority and independent countdown operation?

RQ3. What do the available campus demonstration responses suggest about the appeal of the combined physical setup and lighting system, and what limitations prevent this feedback from establishing the isolated effect of lighting on spectator engagement or player attraction?

The current evaluation does not justify a causal hypothesis test about the isolated effect of the LEDs. A future controlled study may examine whether synchronized semantic lighting increases spectators' event-recognition accuracy or willingness to approach the VR installation relative to a matched condition without event-linked lighting. Such outcomes remain research propositions rather than findings from the present questionnaire.

1.7 Scope, assumptions and limitations

The implemented technical scope comprises a 460-LED external lighting installation, a Unity-to-Arduino communication pathway and three gameplay-linked feedback categories: flight-speed variation, gold ring passing and a movement-triggered 60-second countdown. The four 100-LED main strips display the same synchronized animation; a dedicated 60-LED strip provides temporal feedback. When a ring is passed, its animation takes temporary priority over the speed effect without stopping the countdown.

The research is restricted to the present version of the VR game and demonstration installation. Danger warnings, boosts, victory effects and other potential events are not claimed as implemented in this version. The countdown reaches zero and all 60 timer LEDs turn off, but the VR game currently continues running: automatic gameplay termination at countdown completion has not yet been implemented. The exact speed-range thresholds, animation durations, software versions and timing performance were not available for verification while preparing this text and are intentionally omitted.

The installed LED strips were reported to receive power through the Arduino Uno's 5 V and GND connections. This power arrangement has not been independently assessed and should not be

treated as safe for a full 460-LED load. A suitably designed external power system and professional wiring inspection are required before further full-installation operation. The research makes no claim of certified electrical safety, measured low latency, commercial readiness or statistically demonstrated spectator comprehension.

1.8 Individual contribution

The individual work concerns the external visual feedback subsystem and its event-mapping approach. The reported contributions are: selecting gameplay states suitable for external display; defining speed, ring-passing and countdown feedback patterns; implementing the Arduino-controlled 460-LED installation; integrating the outputs with actual Unity gameplay events; using a shared data signal to drive four synchronized main strips; maintaining a separately controlled timer strip; and managing event priority between ongoing speed feedback and temporary ring-passing animations.

An additional part of the individual research concerns the interpretation of the campus evaluation for spectator engagement and player attraction. The work identifies what may be concluded from the question about the combined physical setup and lighting, and distinguishes that perception measure from the unmeasured outcomes of actual spectator event recognition and lighting-specific conversion to ride participation.

This thesis does not attribute the design or implementation of the group's motion platform, airflow system, haptic vest, shared VR environment or general flight-control algorithms solely to the author. Final individual-authorship statements should be cross-checked against team records and the implemented project files.

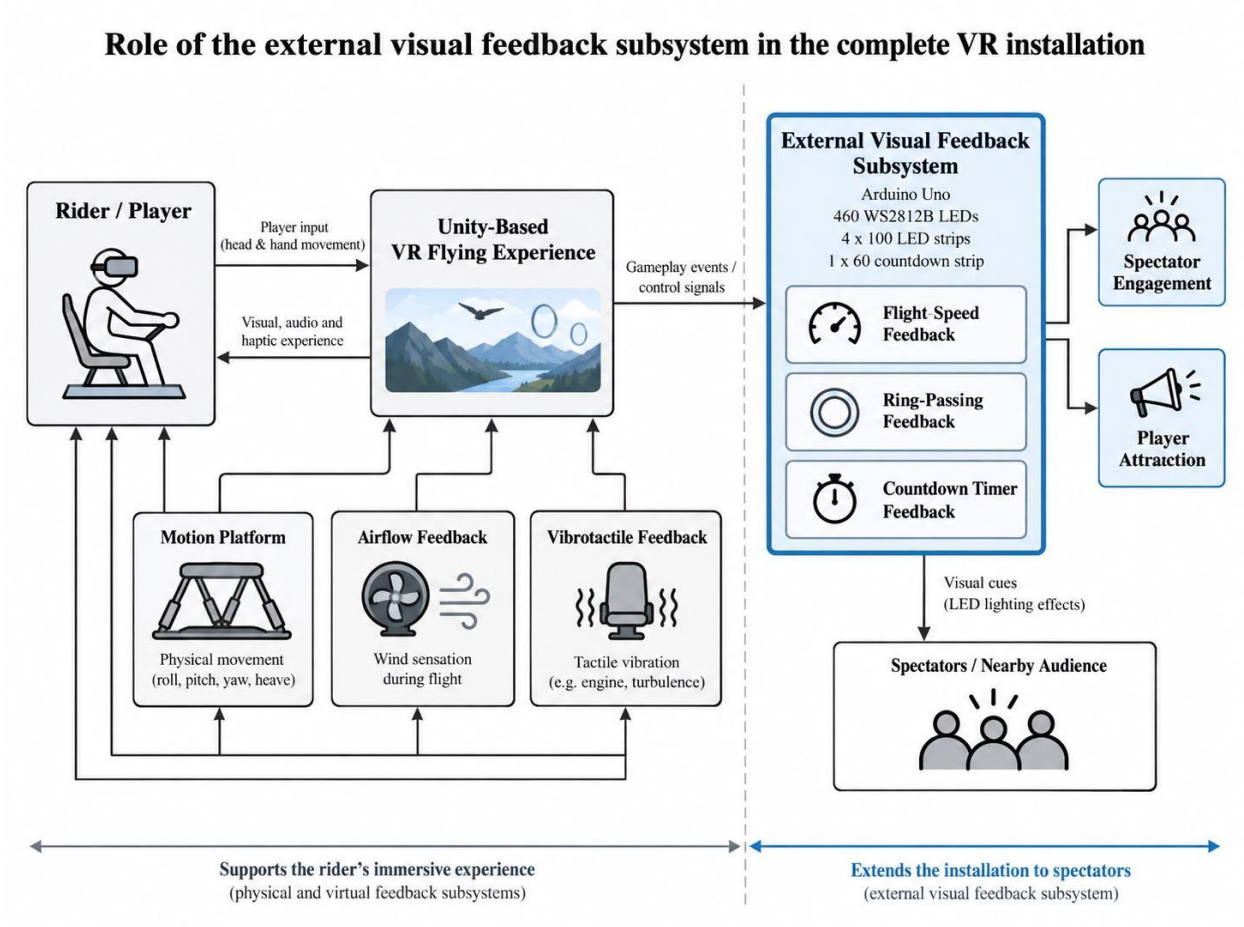


Figure 1.1 Role of the external visual feedback subsystem in the complete VR installation.

1.9 Thesis structure

Chapter 2 reviews the theoretical foundations and relevant approaches to spectator-oriented VR and external visual communication. Chapter 3 describes the design-science approach, the technical demonstration and the available campus questionnaire. Chapter 4 sets out the requirements, architecture and semantic mapping framework. Chapter 5 documents the implemented hardware, software interaction, integration and functional verification. Chapter 6 presents the confirmed results and identifies the limits of the available evidence. Chapter 7 interprets the contributions, compares relevant approaches and discusses limitations and practical implications. Chapter 8 summarizes the work and outlines future technical and research directions.

2 LITERATURE REVIEW AND RELATED WORK

2.1 Review method and boundaries

This preliminary literature review focuses on research relevant to three aspects of the individual component: the difference between immersive and external VR experiences, ways of including spectators in VR activities, and methods for representing information through physical visual cues. It also considers theoretical work on situation awareness and design science as foundations for describing and evaluating the implemented artefact.

The review draws on established scholarly work included in the research material prepared for this draft. It is a narrative review rather than a documented systematic literature review. For the final submission, the author should verify each source against its publisher record, extend the discussion with directly relevant research on ambient information displays and spectator attraction, and record the literature search method and screening criteria. No systematic-search coverage or exhaustive novelty claim is asserted here.

2.2 Domain and theoretical foundations

The experience of the VR rider differs from the experience of a bystander. Steuer describes telepresence in terms of characteristics such as vividness and interactivity [1], while Slater discusses factors contributing to realistic responses inside immersive environments [2]. These foundations help explain why a person wearing an HMD receives a different set of stimuli from someone who can only observe the external equipment. The objective of external lighting is not to reproduce the full perceptual experience of the rider. Rather, it is to make a limited selection of important gameplay information available in a shared physical space.

Endsley's model of situation awareness distinguishes between perceiving information, understanding its meaning and anticipating what may happen next [3]. Applied cautiously to this component, this suggests that a spectator merely noticing a flash does not necessarily understand that a ring was passed. A cue must be sufficiently visible and consistently associated with its intended gameplay meaning. The use of gold lighting for a gold-coloured ring provides one design

rationale for such a relationship, but its actual interpretability to an unfamiliar spectator would need specific evaluation.

Human-centred design principles also emphasize the importance of understanding the intended users and evaluating interactive systems in context [9]. This is relevant because the system has two audiences: spectators who are attempting to follow a ride and potential players who encounter the installation for the first time. These groups may respond differently to the same visual pattern. A highly noticeable effect may help attract attention but may also require explanation before its gameplay meaning becomes apparent.

2.3 Related research

ShareVR, presented by Gugenheimer and colleagues, explored techniques to support co-located participation between HMD and non-HMD users [5]. Its emphasis on allowing additional people to participate in an immersive activity is relevant to the problem of designing VR beyond a single headset wearer. Spectator View, presented by Welsford-Ackroyd and colleagues, investigated asymmetric interaction between HMD wearers and spectators using a larger shared display [6]. Both studies demonstrate the importance of spectator-oriented interaction, although the mechanisms differ from the physical lighting approach used in this project.

These systems communicate through richer visual displays or interaction facilities. The current component instead sends a small set of selected gameplay meanings through an array of LEDs. This deliberately limits the amount of information presented, but it may be appropriate in settings where the spectator is moving around the installation or observing it from a short distance. It also creates the risk that complex events will be reduced to cues that are visually noticeable but semantically ambiguous.

Design Science Research (DSR) provides a useful methodology for developing and evaluating an engineered artefact [4], [7]. The present thesis uses this approach to link the identified spectator-information problem to a designed mapping mechanism, its Unity–Arduino implementation, the observed operation of the completed system and the limitations of its existing evaluation evidence.

2.4 Comparison of spectator communication approaches

Addressable LEDs, Unity and embedded microcontrollers are enabling technologies rather than original inventions of this research. The distinctive design effort lies in choosing which gameplay states warrant external feedback, assigning consistent visual behaviour to those states and managing simultaneous or competing event categories.

Table 2.1 Comparison of spectator communication approaches

Approach	Description stated in the thesis
ShareVR	explored techniques to support co-located participation between HMD and non-HMD users [5].
Spectator View	investigated asymmetric interaction between HMD wearers and spectators using a larger shared display [6].
External LED feedback	The current component instead sends a small set of selected gameplay meanings through an array of LEDs.

2.5 Gap synthesis and conceptual framework

The literature suggests three design requirements. First, information should be selected with the external observer's context in mind, rather than transmitting every internal game variable. Second, different kinds of game information should receive distinguishable visual representations. Third, evaluation should separately consider whether a cue is noticed, whether its meaning is understood and whether it influences interest in participating.

The working framework for this project contains five stages:

1. Gameplay state acquisition: obtaining speed changes, ring-passing triggers and countdown information from the Unity game.

2. State classification: distinguishing continuous, discrete and temporal information.

3. Semantic interpretation: identifying the meaning of each state for someone outside the headset.

4. Visual cue mapping: assigning animation speed, colour, direction, flashing and priority rules.

5. Physical LED output: rendering the assigned patterns on the relevant strips.

The current design encodes continuous motion through animation speed, a discrete gold ring event through a directional gold animation, and temporal progress through sequential LED depletion. Ring feedback temporarily replaces the speed display, whereas countdown information remains on a dedicated strip. This is the conceptual basis for the implemented system and for any future study of spectator comprehension or player attraction.

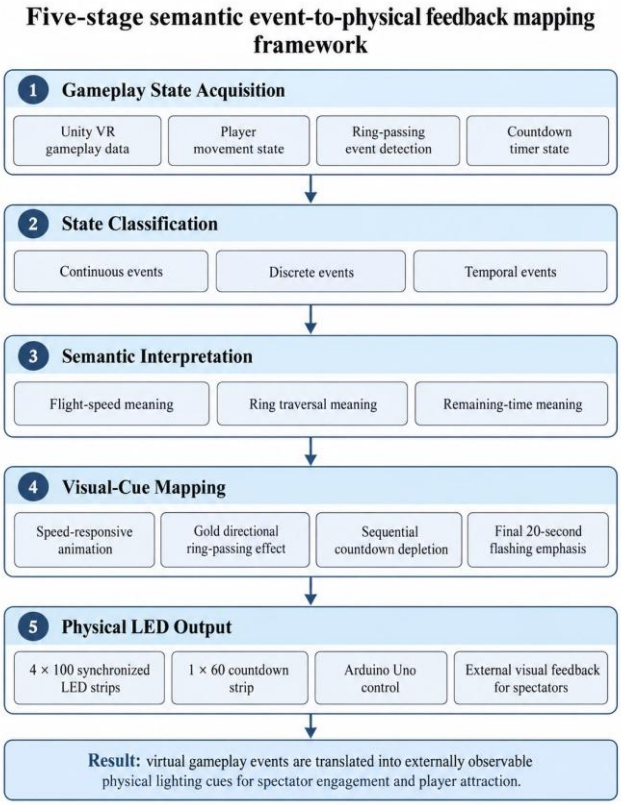


Figure 2.1 Five-stage semantic event-to-physical feedback mapping framework.

3 RESEARCH METHODOLOGY

3.1 Research design

This study adopts a Design Science Research (DSR) orientation, in which a technical artefact is designed to address an identified problem and examined through demonstration and evaluation [4], [7]. The central artefact is a synchronized physical visual feedback subsystem and its semantic mapping approach, implemented as part of an existing multisensory VR installation. The research process comprises problem identification, definition of the individual component objectives, design of a visual mapping scheme, implementation and integration, functional demonstration, and interpretation of the evidence available from the campus evaluation.

Two forms of evaluation must be distinguished. Technical verification concerns whether the lighting responds to gameplay as designed. The participant evaluation concerns attitudes toward the complete physical setup and lighting, rather than the independently measured recognition of game events. The present draft reports these separately because successful implementation does not automatically imply increased spectator comprehension or player attraction.

3.2 Data, participants or study materials

The technical study materials consist of the Unity VR application, the Arduino Uno microcontroller, the 460-LED WS2812B lighting arrangement, the shared-signal output channel for the four main strips, the independent timer channel and the implemented game-event behaviours. The main array includes four 100-LED strips connected to digital Pin 6 for synchronized speed and ring animations. A separate 60-LED strip is connected to digital Pin 7 for the countdown. The project uses actual game-event triggering rather than only a manually generated light show.

The human evaluation was conducted during a university campus demonstration of the complete VR project. A total of 35 people participated in the demonstration. Participants were not divided or labelled as separate rider and spectator groups: the attendees experienced the overall project and were exposed to the external lighting as part of the demonstration. Some viewed the lights

before riding, while others observed them after completing the experience. The questionnaire was designed for the complete group project, not specifically for the LED component.

Of the 35 demonstration participants, 33 answered the single questionnaire item directly relevant to the physical setup and lighting. There were two nonresponses for that item. The source available for this draft is the shared Google Forms response-summary screenshot supplied by the researcher. Its item counts and percentages are reported in Chapter 6. The complete respondent-level export, questionnaire wording for every other project component and the ethics/consent documentation were not supplied for inspection at the time of preparing this draft. They should be retained or referenced in the final research record where appropriate.

3.3 Experimental or study procedure

The completed evaluation was a campus demonstration followed by collection of participant feedback. Attendees encountered the shared multisensory VR installation and its external lighting in the course of the event. There was no independent experimental group exposed only to the LEDs, no documented lighting-on versus lighting-off comparison, and no recorded division between non-riding spectators and VR riders. Accordingly, the procedure is described as a demonstration-based, single-group participant perception evaluation rather than as a controlled spectator experiment.

The system itself was functionally exercised through actual VR gameplay. During normal movement, speed changes altered the animation speed on the four main strips. A ring traversal activated a gold flashing and travelling animation, after which speed feedback resumed. The dedicated countdown began with rider movement, progressed through one LED switching off per second and continued independently when the ring animation was displayed. These observations constitute reported functional testing of implemented behaviour. No timed laboratory benchmark or independent event-logging dataset is asserted.

The group questionnaire included the following statement relevant to the author's component: “The physical setup and lighting made the experience exciting and immersive to approach.”

Participants responded on a five-point agreement scale from Strongly Disagree to Strongly Agree. The response summary provided 33 answers for this item. Because the question combines physical setup and lighting, it measures perception of their combined presentation rather than a lighting-only outcome.

An additional informal observation made during the campus demonstration was that individuals who did not previously know about the project noticed the illuminated installation and approached it with interest; some subsequently tried the experience. The number of people whose decision to participate was attributable to the lights was not recorded. This observation is therefore treated as contextual qualitative feedback, not a quantified measure of conversion or causal attraction.

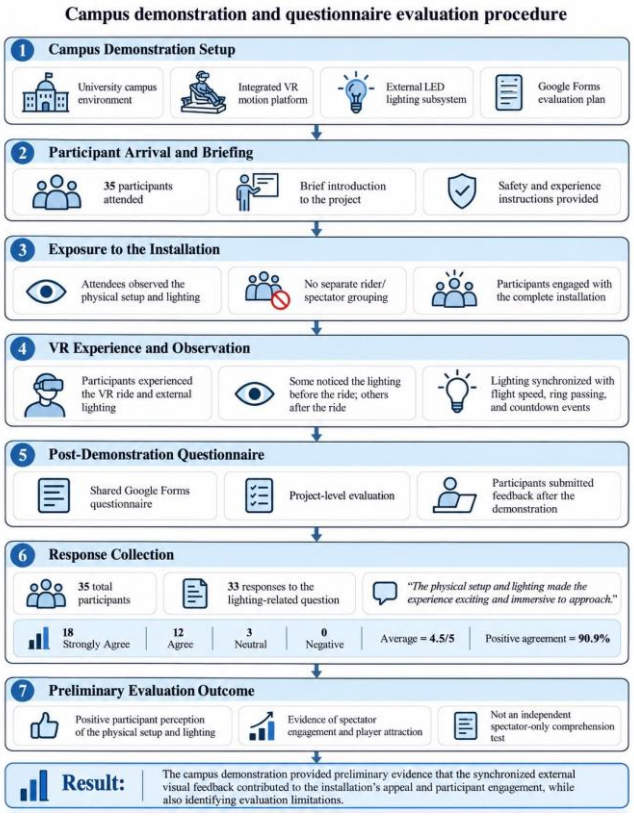


Figure 3.1 Campus demonstration and questionnaire evaluation procedure.

3.4 Measures and evaluation metrics

The current quantitative human measure is agreement with the combined physical-setup and lighting item. The analysis reports the number and proportion selecting each response, the proportion selecting the two positive categories, and the arithmetic mean of the five-point scores as a descriptive indicator. The response denominator is 33, not the 35 people who attended. Two nonresponses are disclosed rather than silently included as neutral or negative answers.

Technical evaluation is currently based on the reported observed operation of the three implemented visual feedback categories, event override behaviour and countdown independence. Variables such as precise event-to-light latency, frame/update timing, packet error rate, memory allocation, LED current draw and operating temperature are not represented by measured results in this draft. They are listed as evidence requirements for a later reproducible technical validation.

3.5 Data analysis

The response distribution was evaluated descriptively. Eighteen participants selected Strongly Agree, twelve selected Agree, and three selected Neutral. No respondent selected Disagree or Strongly Disagree. The positive-response total is therefore 30. Dividing 30 by the 33 respondents yields 90.9% positive agreement. The mean score is calculated as $(18 \times 5 + 12 \times 4 + 3 \times 3) / 33 = 147 / 33$ = approximately 4.45 out of 5, or 4.5 when rounded to one decimal place.

No inferential significance test is appropriate for demonstrating a lighting effect because the available data do not contain a comparison condition or a lighting-specific dependent variable. The research also did not record an objective spectator event-recognition task. This limits conclusions to positive participant perception of the installation as a whole and the reported practical feasibility of the LED mapping mechanism.

The two unanswered questionnaires for this item represent 5.7% of the 35 participants. The analysis uses available-case denominators for the item and does not infer the missing responses. A future evaluation could separate spectator and rider experiences, measure event-recognition

accuracy and compare the installation with and without synchronized lighting using a documented protocol.

3.6 Ethics, privacy, security and responsible AI

Participant feedback must be handled according to the institution's applicable consent and research ethics procedures. For this draft, only aggregate counts from the response-summary image are reproduced; individual identities are neither required nor included. The final report should specify the actual recruitment process, consent arrangements, data-retention rules and any applicable ethics-approval reference after these are confirmed from project records.

Physical safety is an important limitation. A system with 460 WS2812B LEDs may draw substantially more current than an Arduino Uno or typical computer USB connection can safely provide. The researcher reported connecting the LED power wiring to the Arduino Uno 5 V and GND pins. This should not be presented as an approved full-load power architecture. The installation requires an appropriately rated and protected external 5 V distribution system, inspection of wiring and grounds, and safe brightness/thermal assessment before further full-installation use. High-frequency or intense flashing should also be reviewed for viewer comfort and photosensitivity.

Generative AI tools have been used as assistance in structuring and drafting this manuscript. This does not substitute for personal authorship of the technical work, independent verification of claims, appropriate citation of source material or compliance with university policies. The final AI-use declaration should accurately describe the tools and purposes actually involved rather than presenting a generic statement that denies assistance.

3.7 Reproducibility and research artefacts

A reproducible version of this study should preserve the Unity project build, the code responsible for transmitting gameplay events, the Arduino firmware and its LED library dependencies, the event protocol, an exact pinout and independently verified power schematic. It should also include the threshold values used for speed mapping, effect timing, LED brightness levels, countdown activation rules and a dated functional-test log.

The Arduino source and Unity C# files were not available to the researcher while compiling this text. Therefore, the draft records the confirmed observable behaviour and pin assignments without inventing exact code statements, speed thresholds or timing values. For the human evaluation, the questionnaire prompt and the response-summary screenshot should be preserved, and the full anonymized response export added if available. The completed manuscript should cite or append the relevant original evidence wherever permitted by university rules.

4 PRODUCT AND SOLUTION DESIGN

4.1 Requirements

The requirements were derived from the need to display selected gameplay information externally while operating as one subsystem within a shared VR installation. The visual patterns had to be distinguishable by purpose, coordinated with relevant game events and capable of running on the chosen Arduino hardware.

Table 4.1 Requirements traceability for the external feedback component

Requirement stated in Section 4.1	Implementation documented in the thesis
display selected gameplay information externally	The current component instead sends a small set of selected gameplay meanings through an array of LEDs.
distinguishable by purpose	The present design instead selects a small group of gameplay events and gives each an appropriate visual representation.
coordinated with relevant game events	Relevant gameplay information is communicated to the Arduino Uno, which selects the visual output according to the currently active state or event.
capable of running on the chosen Arduino hardware	The Arduino Uno was retained as the working controller despite its memory constraints.

4.2 Component architecture within the complete solution

The data-control path begins in the Unity VR game, where movement values and event triggers are available to the external feedback subsystem. Relevant gameplay information is communicated to the Arduino Uno, which selects the visual output according to the currently active state or event. The Arduino sends the corresponding addressable LED data to two output channels.

The first channel uses digital Pin 6 to distribute one shared data signal to the four 100-LED strips. All four strips show the same animation at the same time. The second channel uses digital Pin 7 to operate the separate 60-LED countdown strip. The two output groups are functionally distinct: a ring event changes the current effect on the main four strips but does not stop or reset the countdown.

The decision to use one shared signal for the main strips arose from memory constraints encountered during development with an Arduino Uno. Sharing a 100-pixel animation sequence across four strips can avoid the need for four independently maintained colour arrays in an implementation that writes identical data to each strip. The student's account indicates that this strategy resolved the encountered memory limitation. Because the final source code and compiled-memory report are unavailable, the exact memory saved is not reported as a measured result.

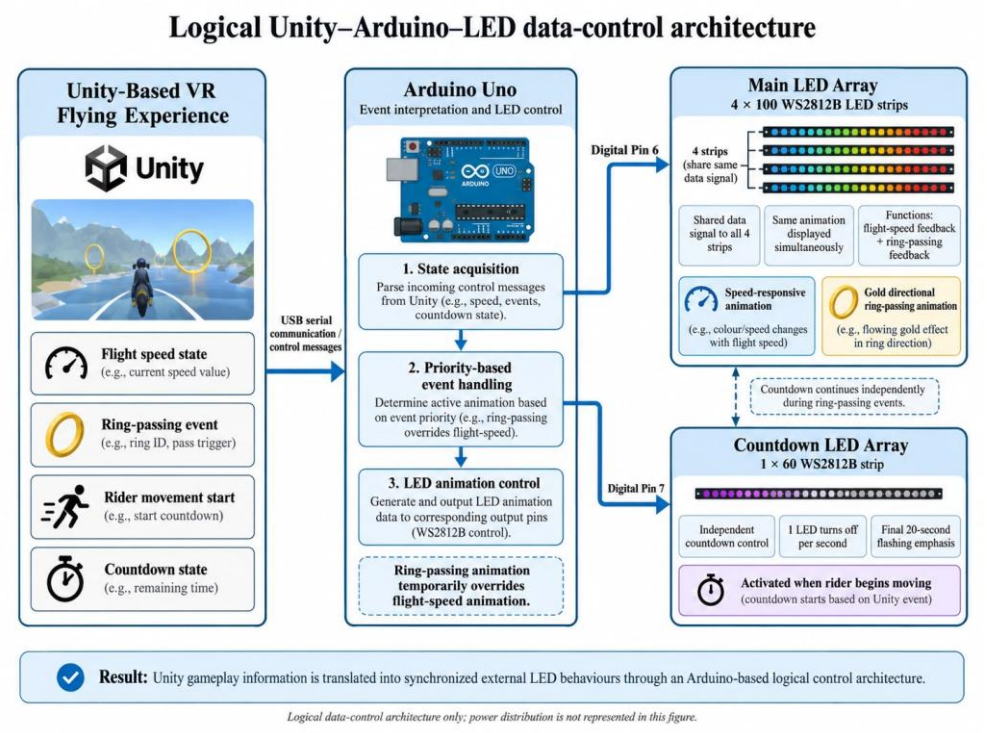


Figure 4.1 Logical Unity–Arduino–LED data-control architecture.

This is a logical data architecture, not a verified power wiring diagram. The electrical power connections require separate documentation and safety review.

4.3 Design alternatives and decisions

A direct representation of all gameplay content on an external display would require a different kind of spectator interface. The present design instead selects a small group of gameplay events and gives each an appropriate visual representation. A persistent base animation is used for

continuous speed information, while a discrete ring event temporarily takes over the main light display. The countdown is assigned a separate physical channel so that time information is not lost during short event effects.

The Arduino Uno was retained as the working controller despite its memory constraints. The reported shared-signal approach allowed the four main strips to be synchronized without driving each strip as an independent display. An ESP32 or another controller with greater processing and memory resources is a potential future alternative; it is not the controller used in the completed implementation.

Colour and movement were combined for the ring-passing effect. The gold hue matches the appearance of the rings within the VR game, while the directional motion suggests passing through an object. A flashing component adds salience. These are reasoned design decisions; the present survey does not establish which element is responsible for event recognition. The final countdown phase adds flashing to the progressively shrinking set of illuminated timer LEDs, visually distinguishing the last 20 seconds from the earlier countdown period.

4.4 Detailed design

The designed event mapping can be expressed through a five-stage sequence: acquisition, classification, semantic interpretation, mapping and physical output. The implemented states are grouped according to their time-related properties. Flight speed is continuous; ring passing is discrete; the countdown is temporal. Their physical encodings are outlined in Table 4.2.

Table 4.2 Gameplay-event to physical visual-feedback mapping

Gameplay information	Classification	Physical feedback stated in the thesis
Flight speed	Continuous	The animation speed changes according to predefined ranges of the virtual flight-speed value.
Ring passing	Discrete	Successful traversal of a gold-coloured ring produces an event-triggered gold flashing animation. The effect

		also includes a visual movement travelling along each main strip from one end to the other.
Countdown	Temporal	Every second, one LED turns off. During the final 20 seconds, the remaining LEDs flash while continuing to extinguish at one-second intervals.

The continuous and discrete effects share the four main strips. During ordinary flight, the LED animation responds to predefined speed ranges. The values defining those ranges have not been made available and must not be reconstructed from examples. When a ring is crossed, the ring animation temporarily overrides speed feedback. Once the ring animation has completed, the speed-based pattern returns. This is a simple event-priority model that prevents the continuous visual update from immediately obscuring a short, meaningful event.

The countdown has its own activation condition. It begins when rider movement begins, uses the fixed one-minute configuration and continues even if ring feedback is being shown on the main strips. The first part of the countdown uses sequential LED extinction; the last 20 seconds add a flashing behaviour to the remaining LEDs. At zero, the timer LEDs are extinguished, but automatic termination of the Unity game remains incomplete. Longer or shorter timer durations may require changes to the existing mapping logic.

In this architecture, four physical strips are synchronized by a shared data line. This must not be confused with reducing the total number of powered LEDs. Although the data sequences may be reused, all 460 physical LEDs remain electrical loads. No exact pixel-update rate, effect duration or end-to-end synchronization tolerance is claimed without access to the final code and measurements.

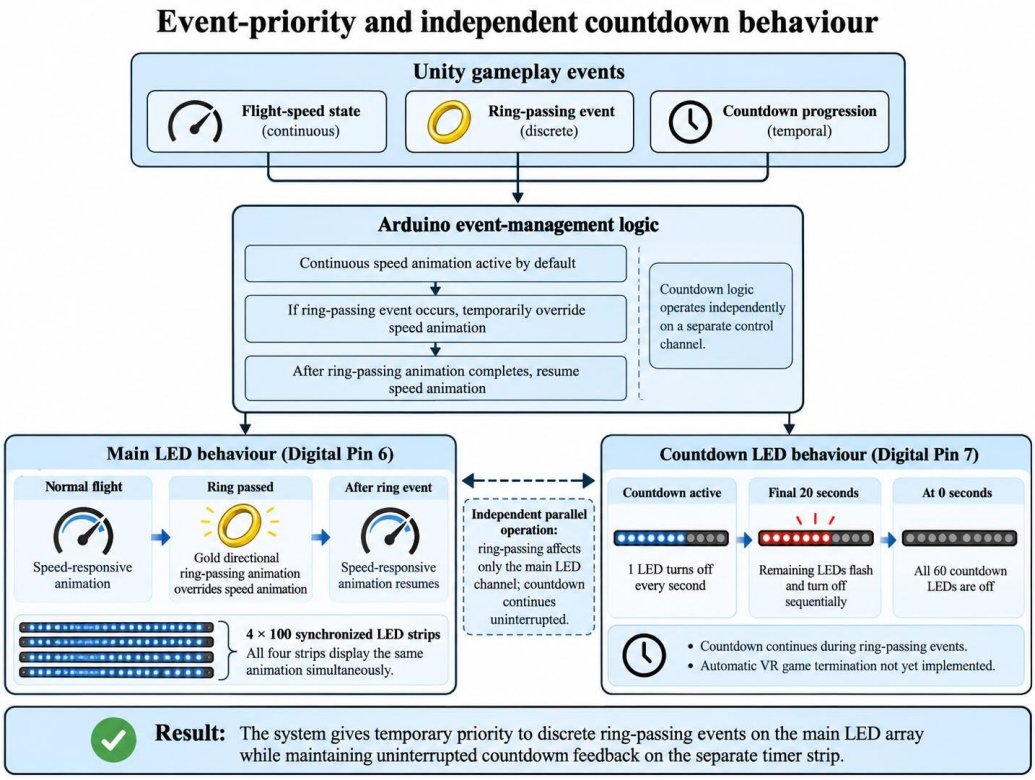


Figure 4.2 Event-priority and independent countdown behaviour.

4.5 Standards and professional practice

Professional implementation of an external lighting system must address more than visual correctness. Power distribution, safe wiring, brightness, viewer comfort, software robustness and maintainability are relevant to an installation used around members of the public. Human-centred design principles require cues to be evaluated from the perspective of intended spectators rather than only by the developer [9].

The reported arrangement of connecting the LED 5 V and GND power leads directly to the Arduino Uno must be independently inspected and corrected before any further full-array operation. A 460-pixel WS2812B installation can, in a hypothetical all-white maximum-brightness condition at 60 mA per pixel, require up to approximately 27.6 A. This is an illustrative electrical sizing calculation, not a measurement of the implemented system. An Uno powered through a typical USB connection is not a suitable supply for that potential load. The final installation requires an appropriately rated external regulated supply, safe distribution paths,

protective measures and a common ground arrangement where appropriate. The project should not be described as electrically certified or safely powered until those requirements have been met and documented.

The system should also be assessed for excessive strobing, glare, viewing-angle limitations and the use of colour as the sole distinction between messages. The ring-passing cue combines hue, flashing and movement, but whether that is accessible and immediately meaningful to different spectators is not known from the available survey. Any future venue deployment would require additional safety, usability and reliability tests.

5 IMPLEMENTATION, INTEGRATION AND TESTING

5.1 Development environment and technology stack

The visual feedback component is integrated with a Unity-based VR game and implemented using an Arduino Uno microcontroller and WS2812B addressable LED strips. Unity supplies gameplay state information and event triggers for the physical feedback. The Arduino operates two logical data output channels: a shared channel for the four 100-LED main strips and a separate channel for the 60-LED countdown strip.

The precise Unity version, Arduino IDE version, firmware revision, serial parameters, library versions and build identifiers were not available for verification. These should be added from the final project files rather than guessed. The absence of exact version information does not invalidate the reported functionality, but it limits reproducibility until supporting artefacts are collected.

5.2 Component implementation

The development process required translation of gameplay events into physical lighting patterns and coordination of those patterns on a resource-constrained microcontroller. The implemented system responds to all the special-event types presently available in the current VR game: flight-speed changes, ring passing and countdown progress.

Flight-speed mapping. During normal gameplay, the speed-responsive LED animation is active on the four main strips. The animation speed changes according to predefined ranges of the virtual flight-speed value. A higher configured range produces a faster moving animation than a lower configured range. The source thresholds and precise update rules are unavailable in this draft and are not represented by illustrative numbers. Functional testing with the Unity game confirmed to the researcher that the feedback changes when the virtual flight speed changes.

Ring-passing mapping. Successful traversal of a gold-coloured ring produces an event-triggered gold flashing animation. The effect also includes a visual movement travelling along each main strip from one end to the other. The chosen hue matches the visual identity of the game rings, while the directional animation is intended to depict the passing action. This is a discrete cue

rather than a permanent animation mode. When it is triggered, it temporarily supersedes the flight-speed pattern; the previous speed-responsive display then resumes after the ring effect ends.

Countdown mapping. The dedicated 60-LED strip operates with the present 60-second game configuration. Its countdown begins when the rider starts moving. Every second, one LED turns off. During the final 20 seconds, the remaining LEDs flash while continuing to extinguish at one-second intervals. The timer reaches a state with all 60 LEDs off. At this stage of the shared game development, the VR session does not automatically end at timer zero, even though automatic session termination is a planned requirement. The timer representation may need modification if the game's countdown duration is changed.

Shared-signal implementation. The completed main-lighting array comprises four strips of 100 pixels that receive the same data signal from Arduino Pin 6. The four strips present the same animations simultaneously. The developer adopted this design after encountering memory constraints when handling multiple strips on the Arduino Uno. The shared design allows repeated presentation of the same logical 100-pixel pattern. Its exact memory use and firmware data structures cannot be quantified without inspecting the actual program. The dedicated timer strip uses Pin 7 and remains separately addressable.

Table 5.1 Implemented functions and documented limitations

Implemented function	Description recorded in the thesis
Flight-speed mapping	The animation speed changes according to predefined ranges of the virtual flight-speed value.
Ring-passing mapping	Successful traversal of a gold-coloured ring produces an event-triggered gold flashing animation. The effect also includes a visual movement travelling along each main strip from one end to the other.
Countdown mapping	Its countdown begins when the rider starts moving. Every second, one LED turns off. During the final 20 seconds, the remaining LEDs flash while continuing to extinguish at one-second intervals.
Shared signal implementation	The completed main-lighting array comprises four strips of 100 pixels that receive the same data signal from Arduino Pin 6. The dedicated timer strip uses Pin 7 and remains separately addressable.

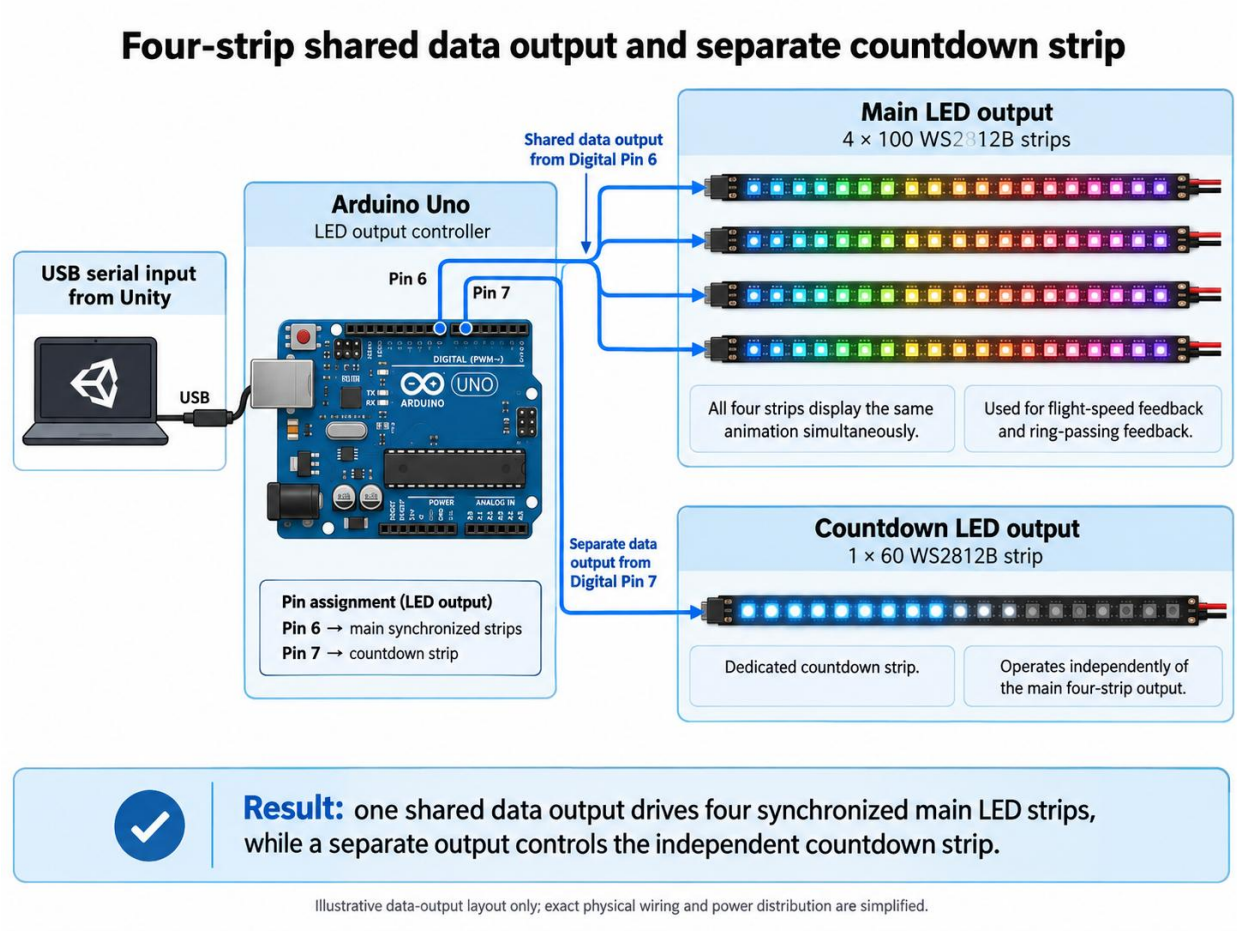


Figure 5.1 Four-strip shared data output and separate countdown strip (logical data layout).

The source code was not available when this draft was compiled; therefore, no undocumented functions, serial message formats, fixed effect durations or arbitrary thresholds are presented as verified implementation facts.

5.3 Integration with the other components

The completed LED subsystem operates alongside the group's VR navigation, mechanical motion, airflow and haptic subsystems. It consumes gameplay information from the shared Unity environment but does not independently control the rider's seat movement, flight physics or wearable feedback. This interface helps maintain a clear boundary between the author's work and the technical work of other team members.

The researcher confirmed that the completed 460-LED installation was tested with real Unity VR gameplay events automatically triggering the lighting system. In particular, speed variations affected animation behaviour, ring traversal produced the event animation, and movement initiated the temporal countdown. The main and timer strips continued to serve their separate roles while the game ran.

An important integration decision was the handling of simultaneous information. During an active ring-passing animation, the main strips give temporary priority to the discrete cue. This does not stop countdown progress on Pin 7. The timer therefore continues to display the passage of time while the main display communicates the ring event. After the discrete effect is complete, the speed-based effect returns. This behaviour is part of the implemented integration and demonstrates the separation of continuous, discrete and temporal information at the output level.

The automatic termination of VR gameplay at the end of the one-minute timer is an integration task that remains incomplete. This limitation belongs to the present state of the shared game and should not be presented as a failure of the independently functioning countdown LEDs.

5.4 Verification and software testing

The technical work was reported as implemented and functionally tested with the actual game. This establishes the presence of the feedback behaviours described above, but the project files and dated test logs were unavailable for independent review during the preparation of the manuscript. Table 5.2 therefore separates the reported observation from the documentary evidence still required for final reproducibility.

Table 5.2 Functional verification and outstanding evidence

Verification aspect	Recorded statement from the thesis
Functional status	The technical work was reported as implemented and functionally tested with the actual game.
Supporting records	the project files and dated test logs were unavailable for independent review during the preparation of the manuscript.
Measurement limits	no claim of a particular latency, packet-loss rate, memory saving, full-load electrical safety or prolonged stability is made.
Further verification	Further verification should retain the Unity build, Arduino firmware, power schematic and dated video or logs demonstrating each test condition.

The functional checks suggest that the system represents the three intended categories of gameplay information. However, no claim of a particular latency, packet-loss rate, memory saving, full-load electrical safety or prolonged stability is made. Further verification should retain the Unity build, Arduino firmware, power schematic and dated video or logs demonstrating each test condition.

5.5 Deployment and operation

The completed lighting array was incorporated into the campus demonstration of the multisensory VR project. The external effects were visible as part of the physical installation, and the researcher observed that the lighting attracted the attention of some attendees who were unfamiliar with the project. Participants also interacted with the complete VR system and later provided feedback through the shared questionnaire.

For any repeat demonstration, the correct operational sequence must include confirmation of an appropriately rated and protected LED supply, safe physical mounting, separation of power and data wiring, and a check of visual comfort. The wiring arrangement reported for the completed prototype has not met this verification requirement. The full array should not be powered again from the Arduino Uno's 5 V pin or a typical USB connection before the power system is reviewed and corrected by a competent person. Once safe operation is established, gameplay events and light responses should be demonstrated using a documented test checklist.

The use of an ESP32 is a possible future improvement, but the controller actually used for this implementation is the Arduino Uno. Any controller or power-system change will require new integration and verification work.

6 RESULTS AND VALIDATION

6.1 Research results

The present evaluation provides two kinds of results. First, the researcher reports successful functional operation of the completed 460-LED system with real Unity gameplay events. Second, the university campus demonstration generated questionnaire data relating to the perceived appeal of the combined physical setup and lighting. These findings do not measure the same construct and should not be combined into a claim that synchronized LEDs independently improved spectator comprehension.

Thirty-five people participated in the campus demonstration. The relevant Google Forms item, “The physical setup and lighting made the experience exciting and immersive to approach,” received 33 answers. Among these respondents, 18 selected Strongly Agree, 12 selected Agree, and three selected Neutral. There were no negative ratings. The positive-response proportion was 30/33, or 90.9%, and the mean score was approximately 4.45 out of 5.

These results indicate that respondents generally perceived the complete installation and its lighting favourably. The single item does not establish how much of the favourable perception was caused by the lighting rather than by the physical rig, rider motion or other aspects of the attraction. It also did not require respondents to correctly identify a speed change, ring passage or countdown event. Consequently, the findings are relevant to the overall experience's initial appeal and perceived immersion, but they cannot independently validate the intended semantic communication function.

Informal observations during the demonstration suggest that some people who had no prior knowledge of the project were drawn toward the lighting installation and later expressed an interest in riding. Because the researcher did not record the number of people who approached specifically because of the LEDs, or compare the event with a lighting-disabled condition, this observation remains qualitative and exploratory.

6.2 Product and technical results

The primary product result is the completed implementation of five LED strips containing a total of 460 WS2812B LEDs. Four 100-LED strips display synchronized patterns from a shared data output, while a separate 60-LED strip depicts the one-minute timer. The subsystem is driven by an Arduino Uno and reacts to actual gameplay information supplied by the Unity environment.

The speed feedback changes animation pace as flight speed changes through the configured ranges. The ring-passing effect uses a gold flashing and directional travelling animation, temporarily replacing the continuous speed display. The countdown begins when the rider starts moving, decreases by one illuminated LED per second and introduces flashing during the final 20 seconds. The countdown display operates independently during ring animations, and it reaches an all-off condition at zero.

The completed game currently continues operating after the countdown LEDs reach zero. Automatic termination is an unimplemented game-integration requirement rather than an accomplished feature. Similarly, the claimed memory improvement from reusing a shared control signal is a plausible and reported implementation outcome, but no compiled-memory or RAM-usage comparison was supplied. The firmware, serial timing and electrical characteristics require further documentary and measurement evidence.

6.3 User or stakeholder validation

Table 6.1 presents the complete available response distribution for the single project questionnaire item that explicitly refers to lighting.

Table 6.1 Participant survey responses

Question: "The physical setup and lighting made the experience exciting and immersive to approach."

Rating	Response	Count	Valid responses (%)
1	Strongly Disagree	0	0.0
2	Disagree	0	0.0
3	Neutral	3	9.1
4	Agree	12	36.4
5	Strongly Agree	18	54.5
	Total	**33**	**100.0**

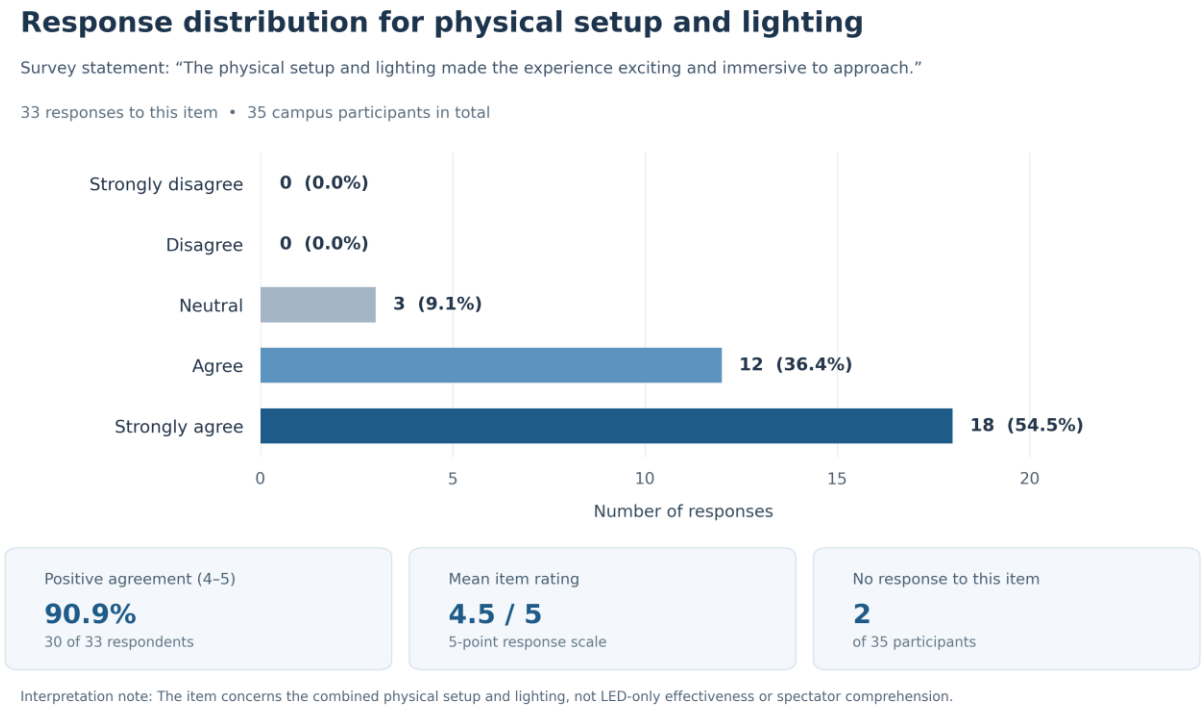


Figure 6.1 Response distribution for the combined physical-setup and lighting survey item.

The count-based values above have been checked arithmetically against the supplied survey screenshot. Figure 6.1 presents the response counts of 0, 0, 3, 12 and 18 for ratings 1 to 5, respectively. The chart uses n = 33 valid item responses, rather than the full 35 campus demonstration participants.

The high positive agreement suggests that the physical presentation of the project was well received by the group that answered this question. The observed approach behaviour of previously unfamiliar attendees supports the relevance of player attraction as a design concern, but does not provide a reliable numerical attraction rate. The measured outcome is combined physical-setup-and-lighting perception, not independent LED effectiveness, and this qualification must remain visible in the abstract and discussion.

6.4 Integration results

The reported functional integration demonstrates that a Unity VR game can trigger event-specific physical LED patterns through the implemented Arduino Uno system. Integration was verified by the researcher using actual gameplay events rather than only standalone test patterns. Continuous movement feedback, ring-triggered directional lighting and temporal countdown output could be used within the same VR experience.

The two-output arrangement proved particularly relevant to simultaneous activity. A ring-passing event temporarily changes what is displayed on the four main strips without interrupting the timer. The countdown remains active on its dedicated data output. Once the ring animation finishes, normal speed feedback resumes. The implementation thus maintains the distinct roles assigned to continuous, discrete and temporal states.

The current integration is nevertheless incomplete in one respect: the gameplay session does not stop automatically when the displayed countdown reaches zero. Also, no timestamped event-to-light test or electrical load certificate is available. These issues are appropriately treated as remaining technical and validation tasks rather than ignored or presented as successful measurements.

6.5 Summary of findings

The present work establishes, on the basis of the researcher's confirmed development history, the functional feasibility of converting selected Unity gameplay states into synchronized, externally visible LED patterns across a complete 460-LED array. The campus evaluation indicates positive participant perceptions of the combined installation and lighting, with 90.9% agreement among the 33 respondents to the relevant question. The observations are consistent with the design goal of producing a visually engaging physical VR installation.

The evidence does not establish a measured improvement in spectator comprehension, nor does it isolate the effect of lighting on the decision to attempt the ride. Quantitative synchronization, memory use, electrical loading and long-term reliability remain undocumented. These distinctions are central to an academically defensible interpretation of the results.

7 DISCUSSION

7.1 Interpretation of findings

The external visual feedback system demonstrates a practical distinction between displaying a visually appealing animation and representing selected game information. Flight-speed variation is expressed through changing animation pace, a successful ring passage is marked by a gold travelling effect, and remaining ride time is expressed through sequential light depletion. Each design corresponds to a different relationship between event duration, meaning and physical representation.

The ring-passing pattern provides an especially clear example of a deliberate semantic design choice. Gold was selected because the target rings are gold in the VR environment, while movement along the strip is intended to represent passage through a target. The animation temporarily takes priority over ordinary speed feedback, thereby giving a short event time to become visible. At the same time, the independently controlled countdown can continue communicating elapsed time. These decisions make the subsystem more structured than a collection of unrelated reactive effects.

The available campus survey result is encouraging but narrow. A 90.9% positive agreement rate relates to the entire physical setup and lighting considered together. This is relevant to the appeal of the installation, yet no separate lighting condition or objective recognition test was used. The results therefore support a conclusion of favourable overall participant perception rather than proven spectator comprehension or a measured causal effect on player attraction.

7.2 Comparison with literature, baselines and existing solutions

Studies such as ShareVR and Spectator View have addressed the challenge of including people who are not wearing the VR headset [5], [6]. Their solutions tend to communicate through more detailed scene views, projection or interactive interfaces. The component developed in this project uses a different level of information representation. It deliberately compresses gameplay into a limited visual vocabulary displayed through physical lights.

The benefit of this approach is potentially greater visibility from the space around an installation without requiring spectators to inspect detailed on-screen information. It may also contribute to the presentation of the installation as an attraction. However, light-based communication cannot convey the spatial and narrative detail of a spectator monitor, and there is a risk that a spectator will notice movement or flashing without understanding the intended event. The current evaluation cannot determine the extent of this trade-off.

An appropriate future baseline would present the same VR experience with otherwise identical physical setup and spectator viewing arrangements, but without gameplay-linked external LED cues. A second condition would add the synchronized mapping system. Such a comparison could measure both event recognition and attraction-related behaviour while holding other factors constant.

7.3 Research contribution

The main research contribution is a structured approach to translating gameplay state into physical feedback according to the type and meaning of an event. The framework organizes selected states into continuous, discrete and temporal categories and assigns a different physical encoding to each. It also addresses the need for temporary event priority and concurrent timing information.

The completed component demonstrates how this framework can be applied to a real Unity-based VR installation with limited embedded hardware resources. Its application provides a design case for considering external lighting as an information channel for spectators and a visible part of a potential participant's first encounter with the attraction. The research contribution is best understood as the proposed mapping framework, its implemented design decisions and the limitations identified for rigorous subsequent evaluation.

The framework is not yet a validated general theory of spectator comprehension or attraction. The current questionnaire cannot show whether a participant understood the ring effect or whether they approached solely because of the lights. Those questions require additional instruments designed specifically for the individual research component.

7.4 Product contribution

The product contribution is an integrated 460-LED visual feedback subsystem using an Arduino Uno and event-driven input from the Unity VR application. Four 100-LED strips display identical

animations through a shared data line on Pin 6, and a separate 60-LED strip displays the countdown through Pin 7. The component responds to normal flight speed, ring traversal and temporal progress while maintaining a meaningful event-override relationship.

The developer's shared-signal design addresses an encountered limitation in microcontroller memory usage, but no numerical RAM saving is claimed. The resulting product is a functional research installation rather than a commercially certified unit. The power supply arrangement reported during drafting must be corrected and validated before it can be considered operationally safe for further public demonstrations.

7.5 Limitations and threats to validity

The evaluation has important limitations. All 35 demonstration participants formed one mixed group; individuals were not separated into rider and spectator categories. The only component-relevant questionnaire item combined physical setup and lighting into a single statement. Two attendees did not answer that item. There was no control condition, no lighting-disabled comparison and no objective test of the respondents' understanding of gameplay events. The group survey was not designed to isolate this individual's subsystem.

The user observation that unfamiliar people approached after noticing the lighting has value as preliminary field evidence but was not captured in a standardized count, interview or observation protocol. Without a comparison and independently recorded outcomes, it would be inappropriate to describe this as a demonstrated increase in player participation caused by the LED system.

The current game includes only the three implemented feedback categories described in this thesis. Its automatic termination at timer zero remains incomplete. The timer configuration is one minute, and changes to the duration may require alterations to the mapping. The speed thresholds, firmware revision and effect timing cannot be reproduced from the available material. End-to-end latency, serial reliability, electrical demand and thermal behaviour also lack quantitative measurements.

A serious technical limitation concerns electrical power. The 460-LED wiring was reported as receiving power through the Arduino Uno 5 V/GND connections. That arrangement is not a demonstrated safe power-distribution design for such a large LED array. The safety concerns must be resolved and documented, independently of whether the animation behaviour appeared to function during a limited demonstration.

7.6 Implications

The findings suggest that low-resolution physical visual feedback may serve two complementary purposes in VR installations. First, selected event semantics can be made externally visible without reproducing the full HMD view. Second, an active physical display may contribute to an installation's visual appeal, which is relevant to exhibition and location-based entertainment contexts. The present study supports the design feasibility of the first function and provides preliminary perception evidence related to the second.

Future practical applications could include VR arcades, educational demonstrations, museums and exhibitions, provided that mapping rules are adapted to the relevant experience and validated with its intended spectators. A commercially deployable system would require safe electrical design, documented maintenance procedures, repeatable performance measurements and evaluation of visitor behaviour. These are potential applications, not proven market outcomes.

The modular logic also suggests future development opportunities: additional event categories, configurable timer lengths, brightness adjustments, accessible cue patterns and a controller with additional processing resources. Any extension should preserve the difference between making an event visible and demonstrating that its meaning is understood by an observer.

8 CONCLUSION AND FUTURE DIRECTIONS

8.1 Conclusion

This research designed and implemented the individual component titled “SYNCHRONIZED EXTERNAL VISUAL FEEDBACK FOR SPECTATOR ENGAGEMENT & PLAYER ATTRACTION” as part of the Immersive Multisensory VR Motion System for Full Body Navigation. The completed installation comprises 460 WS2812B LEDs controlled by an Arduino Uno and linked to real gameplay activity in Unity. Four synchronized 100-LED strips communicate flight-speed variation and ring passage, while a separate 60-LED strip communicates a movement-triggered one-minute countdown.

The implementation applies a semantic event-to-physical feedback approach that distinguishes continuous, discrete and temporal game information. It combines range-based animation-speed feedback, a gold flashing directional effect for ring passage, event-priority handling and independent countdown operation. The developer reported that the system was implemented and functionally tested during actual gameplay.

The campus evaluation involved 35 participants, with 33 valid answers to a question about the combined physical setup and lighting. Thirty of these respondents, or 90.9%, gave positive responses. This result supports a cautious conclusion that the overall installation and its lighting were perceived favourably by the respondents. It does not demonstrate that lighting alone improved spectator comprehension or independently attracted a measurable number of new players. The thesis therefore presents a functional research artefact and preliminary engagement-related evidence, alongside explicit limitations concerning evaluation and technical verification.

8.2 Achievement of objectives

O1 - Identification of relevant game states: Achieved for the current implementation. Flight-speed changes, ring passing and one-minute countdown were identified as the available types of gameplay information relevant to the external display.

O2 - Semantic mapping and event coordination: Achieved at the functional design level. Distinct mappings were defined for continuous, discrete and temporal events, with ring-passing priority and independent countdown operation.

O3 - Implementation and Unity integration: Achieved as reported by the researcher. The complete 460-LED array was implemented and triggered from actual Unity gameplay using an Arduino Uno and the confirmed Pin 6 / Pin 7 arrangement.

O4 - Technical verification: Partially achieved. The relevant functional behaviours were reported and tested, but quantitative latency, repeatability logs, measured memory usage and full-array electrical safety verification were not available.

O5 - Evaluation of spectator engagement and player attraction: Partially achieved. The combined installation received favorable questionnaire responses, and attraction-related informal observations were recorded. Isolated lighting effects, event-recognition accuracy and changes in participation behaviour were not measured.

8.3 Individual research and product contributions

Research contribution. The study presents a five-stage semantic event-to-physical mapping framework, a division of gameplay information into continuous, discrete and temporal categories, a practical priority scheme for competing output events and a clear assessment of the evidence required to evaluate spectator engagement and player attraction separately. This contribution is tied to the implemented VR demonstration, not asserted as a universally validated model.

Product contribution. The author implemented a 460-LED subsystem, synchronized four 100-LED strips through a shared Arduino Uno data signal on Pin 6, and used Pin 7 for the independent 60-LED timer strip. The lighting system responds to Unity flight-speed changes, gold-ring passage and countdown progress. The event-driven main display and independent countdown demonstrate the functional coordination of multiple physical visual cues in the shared VR project.

The individual's authorship concerns the external lighting design, implementation and research analysis. Components developed by the other group members remain recognized as collaborative project context.

8.4 Future directions

The immediate technical priority is to review and redesign the electrical power architecture so the LED strips receive power from a suitably rated and protected external supply rather than relying on the Arduino Uno 5 V connection. The completed installation should then undergo safe-load, temperature, brightness and prolonged-operation tests. Firmware and Unity source files should be

archived, and a repeatable verification report should include event-to-light latency, communication reliability, actual memory use and the exact flight-speed mapping thresholds.

The incomplete automatic termination of the VR game at countdown zero should be implemented and tested. A configurable countdown mapping would also permit durations other than 60 seconds without requiring manual changes to the display logic. Additional gameplay events should only be added once they exist in the shared VR game and can be clearly differentiated from the current cues.

A more rigorous user study should independently investigate spectator event recognition and player attraction. For event recognition, spectators could view matched gameplay sequences with and without event-linked lighting and identify the events observed. For attraction, observations or questionnaires could distinguish who noticed the lights, who approached the installation and who chose to participate. Rider and non-rider roles should be recorded where relevant. Future work may also consider brightness adaptation, colour-vision-accessible patterns and an ESP32-based controller, with any hardware changes subjected to fresh validation.

REFERENCES

[1] J. Steuer, “Defining virtual reality: Dimensions determining telepresence,” Journal of Communication, vol. 42, no. 4, pp. 73–93, 1992. doi: 10.1111/j.1460-2466.1992.tb00812.x.

[2] M. Slater, “Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments,” Philosophical Transactions of the Royal Society B: Biological Sciences, vol. 364, no. 1535, pp. 3549–3557, 2009. doi: 10.1098/rstb.2009.0138.

[3] M. R. Endsley, “Toward a theory of situation awareness in dynamic systems,” Human Factors, vol. 37, no. 1, pp. 32–64, 1995. doi: 10.1518/001872095779049543.

[4] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” MIS Quarterly, vol. 28, no. 1, pp. 75–105, 2004. doi: 10.2307/25148625.

[5] J. Gugenheimer, E. Stemasov, J. Frommel, and E. Rukzio, “ShareVR: Enabling co-located experiences for virtual reality between HMD and non-HMD users,” in Proceedings of the ACM CHI Conference on Human Factors in Computing Systems, 2017. doi: 10.1145/3025453.3025683.

[6] F. Welsford-Ackroyd, A. Chalmers, R. K. dos Anjos, D. Medeiros, H. Kim, and T.-h. Rhee, “Spectator View: Enabling asymmetric interaction between HMD wearers and spectators with a large display,” Proceedings of the ACM on Human-Computer Interaction, vol. 5, no. ISS, art. 485, 2021. doi: 10.1145/3486951.

[7] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” Journal of Management Information Systems, vol. 24, no. 3, pp. 45–77, 2007–2008. doi: 10.2753/MIS0742-1222240302.

[8] A. Bangor, P. T. Kortum, and J. T. Miller, “An empirical evaluation of the System Usability Scale,” International Journal of Human–Computer Interaction, vol. 24, no. 6, pp. 574–594, 2008. doi: 10.1080/10447310802205776.

[9] ISO, ISO 9241-210:2019, Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems, 2019.

[10] FastLED, “Power Management Functions and API Documentation,” https://fastled.io/docs/d3/d1d/group___power.html (accessed October 2026).

[11] United Nations, “Transforming our world: the 2030 Agenda for Sustainable Development,” 2015, https://sdgs.un.org/2030agenda.

[12] G. Goncalves, M. Melo, J. Vasconcelos-Raposo, and M. Bessa, “Impact of Different Sensory Stimuli on Presence in Credible Virtual Environments,” IEEE Transactions on Visualization and Computer Graphics, vol. 26, no. 11, pp. 3231–3240, 2020. doi: 10.1109/TVCG.2019.2926978.

[13] P. Caserman, A. Garcia-Agundez, A. Gámez Zerban, and S. Göbel, “Cybersickness in current-generation virtual reality head-mounted displays: systematic review and outlook,” Virtual Reality, vol. 25, pp. 1153–1170, 2021. doi: 10.1007/s10055-021-00513-6.

GLOSSARY

Addressable LED: A light-emitting diode whose colour can be individually controlled as part of a digital pixel sequence.

Continuous gameplay state: An ongoing or regularly changing game value, such as flight speed, represented through sustained or updated visual output.

Discrete gameplay event: A brief identifiable occurrence, such as passing a target ring, represented through a temporary event-specific pattern.

Event priority: A rule determining which visual feedback appears when different gameplay information types require the same physical display.

External visual feedback: Physical lighting or visual output positioned outside the virtual headset and driven by gameplay-related information.

Player attraction: The interest generated among potential participants in approaching or attempting an interactive installation; this study does not quantify a lighting-only attraction rate.

Semantic event mapping: A design approach that selects a physical visual representation according to the intended meaning and temporal character of a game event.

Spectator engagement: Attention, involvement or interest among observers outside the VR headset. Engagement is conceptually distinct from accurately understanding a game event.

Spectator comprehension: A viewer's ability to correctly interpret selected gameplay states or events from externally available information.

Temporal feedback: Output that represents elapsed or remaining time, such as the progressive extinction of countdown LEDs.

APPENDICES

Appendix A: Campus Questionnaire Item and Result Summary.

Question: “The physical setup and lighting made the experience exciting and immersive to approach.”

Response scale: 1 = Strongly Disagree; 2 = Disagree; 3 = Neutral; 4 = Agree; 5 = Strongly Agree.

Total campus participants: 35.

Valid responses to this question: 33.

Did not answer this question: 2.

Score 1: 0 respondents.

Score 2: 0 respondents.

Score 3: 3 respondents.

Score 4: 12 respondents.

Score 5: 18 respondents.

Positive-response calculation: (12 + 18) / 33 × 100 = 90.9%.

Mean-score calculation: (3 × 3 + 12 × 4 + 18 × 5) / 33 = 147 / 33 = 4.45/5 approximately.

Interpretation limitation: The question evaluated the physical setup and lighting together and did not isolate LED effectiveness, event recognition or conversion from spectator to rider.

Appendix B: Detailed System Specification for Formatting

Research project: IMMERSIVE MULTISENSORY VR MOTION SYSTEM FOR FULL BODY NAVIGATION.

Official individual component title: SYNCHRONIZED EXTERNAL VISUAL FEEDBACK FOR SPECTATOR ENGAGEMENT & PLAYER ATTRACTION.

Student: Nimsara L G S K; IT22191724; Group R26-IM-004.

Supervisor: Mr. Ishara Gamage.

Co-supervisor: Mr. Jeewaka Perera.

Microcontroller: Arduino Uno.

Addressable LED type: WS2812B.

Main display: Four strips containing 100 LEDs each; shared data output Arduino digital Pin 6; identical synchronized animations.

Temporal display: One strip containing 60 LEDs; separate Arduino digital Pin 7.

Total physical pixels: 460.

Implemented speed event: Flight-speed-responsive animation using predefined ranges, exact numeric thresholds unavailable.

Implemented ring event: Gold flashing and a directional travelling animation; temporarily overrides speed feedback, then normal speed feedback resumes.

Implemented countdown: Starts with rider movement; 60 seconds; one LED off per second; final 20 LEDs flash and sequentially extinguish; uninterrupted by ring effects.

Countdown zero state: All timer LEDs off; Unity gameplay continues because automatic end-of-game behaviour is not yet implemented.

Integration: Actual Unity gameplay events automatically trigger the system.

Reported development strategy: A shared data signal to four 100-LED strips was used to address Arduino memory constraints; compiled RAM savings not independently measured.

Power: Researcher reported connecting LED power leads to Arduino Uno 5 V/GND. This requires immediate engineering review and a properly rated external power solution before further operation. Do not portray the reported wiring as a safe recommended circuit.

Exact code, effect timings, firmware build and quantitative test logs: Not supplied during preparation of this draft.

Appendix C: Functional Test Record Template

Record each test as follows:

Test ID | Date | Unity build/version | Arduino firmware version | Input or game action | Expected light output | Observed light output | Pass/Fail | Photo/video/log reference | Notes.

Required documented tests include: changing flight speed; passing a ring; resuming speed after ring animation; countdown initiation on rider movement; timer decrement; final-20-second flashing; ring event during countdown; timer zero; restart conditions; serial disconnect/reconnect; and verified safe loading under an approved external LED supply.

Appendix D: Academic Integrity and Responsible Use of AI

Draft disclosure to revise against the university's actual policy and use history:

“Generative AI assistance was used in the organization, drafting and language refinement of the thesis manuscript. The student provided the project-specific implementation details, individual research scope and participant response information and remains responsible for independently checking the accuracy of the technical descriptions, literature references, calculations, methodology and conclusions. No participant dataset, experimental measurement or statistical significance result was generated for this manuscript. Assistance with manuscript preparation does not constitute independent technical validation.”

[STUDENT: Review and adjust the wording to reflect all AI tools actually used and the exact institutional disclosure policy. Never sign an inaccurate authorship or AI-use statement.]

Appendix E: Evidence Required Before Final Submission

1. Confirm the exact official degree, department and thesis submission requirements against the current institutional instructions.2. Add source code for the final Unity-to-Arduino interface and Arduino firmware, together with accurate software versions and speed threshold values, if available before final submission.

3. Add as-built data wiring and professionally reviewed power diagrams; resolve the unsafe reported direct LED power connection.

4. Insert original photographs of the five LED strips and the physical project installation, with relevant captions and permission.

5. Include the original Google Forms screenshot and, where allowed, the anonymized response export that supports the item distribution.

6. Verify all literature references and cross-check each in-text IEEE citation.

7. Add dated test records, latency measurements, communication reliability and electrical load results only if actually collected.

8. Verify the role of each team member and distinguish shared VR development from this student's individual external-lighting work.

9. Review and sign the student declaration only after completing academic integrity and AI-use requirements.

10. Update automatic Table of Contents, figure/table lists, figure numbers, section numbering and page references in Word.

IMMERSIVE MULTISENSORY VR MOTION SYSTEM FOR FULL BODY NAVIGATION

INDIVIDUAL THESIS

Gunawardena L. H

(IT22082992)

Individual component: Motion Rendering and Control for Embodied VR Navigation

Group ID: R26-IM-004

Report submitted in partial fulfilment of the requirements for the
Bachelor of Science (Hons) in Information Technology

Specializing in Interactive Media

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

October 2026

DECLARATION AND SUPERVISOR CERTIFICATION

Candidate declaration. I declare that this thesis is my own work and does not incorporate, without acknowledgement, material previously submitted for a degree or diploma at any university or institute of higher learning. To the best of my knowledge, it contains no material previously published or written by another person except where acknowledgement is made in the text. I grant the Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute this thesis, in whole or in part, in print, electronic, or other media, while retaining the right to use the content in future work.

Academic integrity and AI disclosure. I have disclosed all use of generative AI and other automated tools in accordance with the module policy. I remain responsible for the accuracy, originality, citations, code, data, analyses, and conclusions in this thesis.

Student name: Gunawardena L H

Student ID: IT22082992

Signature: _____ Date: _____

Supervisor certification. The above candidate carried out this research for the bachelor's degree thesis under my supervision.

Role and name	Signature	Date
Supervisor: Mr. Ishara Gamage		
Co-supervisor: Mr. Jeewaka Perera		

ABSTRACT

Most virtual reality (VR) flying experiences rely on handheld joysticks. This separation between what a user sees visually and what their body physically feels often causes motion sickness, dizziness, and breaks overall immersion. While physical motion platforms help bridge this gap, translating continuous, agile aerial movement onto a compact motion seat is challenging because mechanical actuators have strict tilt limits and can easily bottom out.

This research develops a real-time motion rendering and flight control framework designed for a broom-based, seated VR navigation platform. Following a Design Science Research methodology, an end-to-end control system was created to deliver intuitive full-body navigation. Forward speed and vertical altitude are controlled naturally by squeezing a handle-mounted Force Sensing Resistor (FSR), while directional turns are captured through head-mounted gyroscope tracking. Within Unity, a custom motion rendering engine extracts flight data—including position, velocity, and acceleration—and uses linear interpolation (Mathf.Lerp) smoothing alongside dead zone filtering to prevent unintended steering drift and jerky movement. These computed flight commands are transmitted through a serial communication interface to a microcontroller that drives the physical seat actuators within safe mechanical thresholds.

The system was evaluated through user trials involving 35 participants navigating a flight course. The input mechanics proved highly responsive, with 74.3% of participants rating the pressure-based throttle and altitude response positively, achieving an average rating of 4.2 out of 5. The evaluation also identified key areas for improvement, such as physical grip fatigue over extended sessions and rotational drift during sharp head turns. This individual component contributes the physical grip input pipeline, the motion filtering and rendering algorithms, the embedded actuator control link, and the empirical user validation.

Keywords: VR Locomotion, Motion Rendering, Actuator Control, Force Sensing Resistor, Embodied Interaction

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, co-supervisor, external supervisors and the academic staff of the Faculty of Computing at the Sri Lanka Institute of Information Technology (SLIIT) for their continuous guidance, insightful feedback, and encouragement throughout the design and development of this research project. Their advice was invaluable in refining the motion cueing strategies, control algorithms, and safety parameters of this work.

I am grateful to my team members of Project Group R26-IM-004 for their seamless collaboration, dedication, and technical synergy in constructing and integrating the complete multisensory VR motion platform.

Finally, I extend my heartfelt thanks to the 35 participants who volunteered their time to take part in the experimental trials and user evaluations. Their constructive feedback and engagement made the empirical validation of this flight motion system possible.

TABLE OF CONTENTS

1 INTRODUCTION 1
1.1 Overall project context 1
1.2 Individual component context 2
1.3 Problem statement 3
1.4 Research gap 4
1.5 Aim and objectives 5
1.6 Research questions or hypotheses 6
1.7 Scope limitations and assumptions 6
1.8 Individual contribution 7
1.9 Thesis structure 9
2 LITERATURE REVIEW AND RELATED WORK 10
2.1 Review method and scope 10
2.2 Domain and theoretical foundations 10
2.2.1 Sensory Conflict Theory and Cybersickness 10
2.2.2 Vestibular Motion Perception and Tilt-Coordination 11
2.3 Related research 12
2.3.1 Embodied VR Locomotion Systems 12
2.3.2 Seated Motion Cueing Implementations 12
2.4 Existing systems products and techniques 13
2.5 Gap synthesis and conceptual framework 14
2.5.1 Gap Synthesis 14
2.5.2 Conceptual Framework 15
3 RESEARCH METHODOLOGY 16
3.1 Research design 16
3.2 Data participants or study materials 16
3.2.1 Participant Sample 16
3.2.2 Experimental Materials and Apparatus 17
3.3 Experimental or study procedure 17
3.4 Measures and evaluation metrics 18
3.4.1 System Performance Metrics 18
3.4.2 User Experience Metrics 18
3.5 Data analysis 18
3.6 Ethics privacy security and responsible AI 19

3.7 Reproducibility and research artefacts 20
4 PRODUCT AND SOLUTION DESIGN 21
4.1 Requirements 21
4.1.1 Functional Requirements 21
4.1.2 Non-Functional Requirements 22
4.2 Component architecture within the complete solution 22
4.3 Design alternatives and decisions 24
4.4 Detailed design 25
4.5 Standards and professional practice 25
5 IMPLEMENTATION INTEGRATION AND TESTING 26
5.1 Development environment and technology stack 26
5.2 Component implementation 26
5.3 Integration with the other components 27
5.4 Verification and software testing 27
5.5 Deployment and operation 28
6 RESULTS AND VALIDATION 29
6.1 Research results 29
6.2 Product and technical results 30
6.3 User or stakeholder validation 31
6.4 Integration results 31
6.5 Summary of findings 32
7 DISCUSSION 33
7.1 Interpretation of findings 33
7.2 Comparison with literature baselines and existing solutions 33
7.3 Research contribution 34
7.4 Product contribution 34
7.5 Limitations and threats to validity 35
7.6 Implications 36
8 CONCLUSION AND FUTURE DIRECTIONS 37
8.1 Conclusion 37
8.2 Achievement of objectives 37
8.3 Individual research and product contributions 38
8.3.1 Research Contributions 38
8.3.2 Product Contributions 38
8.4 Future directions 39

REFERENCES 40
GLOSSARY 42
APPENDICES 43
Appendix A Research Instruments or Extended Results 43
A.1 Post-Exposure System Evaluation Questionnaire 43
A.2 Extended User Response Distribution 44
A.3 Telemetry Data Frame Specification 44
Appendix B Academic Integrity and AI Use Disclosure 44
B.1 AI Tools and Usage Scope 44
B.2 Statement of Authorship and Verification 45

LIST OF FIGURES

Figure No.	Title	Page
Figure 1.1	Functional block architecture of the motion rendering and control subsystem (Original in Color)	9
Figure 2.1	Conceptual framework of the motion rendering and control pipeline (Original in Color)	16
Figure 3.1	Sequential experimental trial protocol for human-subject evaluation (Original in Color)	24

LIST OF TABLES

Table 1: Comparison of existing VR locomotion and motion rendering approaches.13
Table 2: Architectural design decisions and trade-off evaluations 24
Table 3: Participant evaluation of handle-grip throttle and altitude responsiveness. 31

LIST OF ABBREVIATIONS

Abbreviation	Meaning
2-DOF	Two Degrees of Freedom
3-DOF	Three Degrees of Freedom
6-DOF	Six Degrees of Freedom
AC	Alternating Current
ADC	Analog-to-Digital Converter
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
B2B	Business-to-Business
CAD	Computer-Aided Design
DC	Direct Current
DSR	Design Science Research
FEA	Finite Element Analysis
FoS	Factor of Safety
FSR	Force Sensing Resistor
HCI	Human-Computer Interaction
HMD	Head-Mounted Display
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
IP	Intellectual Property
ISO	International Organization for Standardization
IT	Information Technology
LBE	Location-Based Entertainment
LED	Light-Emitting Diode
MCU	Microcontroller Unit
MIG	Metal Inert Gas
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
NFR	Non-Functional Requirement
PWM	Pulse Width Modulation
SDG	Sustainable Development Goal
SHS	Square Hollow Section
SLIIT	Sri Lanka Institute of Information Technology
SSQ	Simulator Sickness Questionnaire
UART	Universal Asynchronous Receiver-Transmitter
UI	User Interface
USB	Universal Serial Bus
VR	Virtual Reality
WBS	Work Breakdown Structure
XOR	Exclusive OR

1 INTRODUCTION

1.1 Overall project context

Virtual reality (VR) technologies have advanced substantially over recent years, providing high-resolution head-mounted displays, wide viewing angles, and responsive optical tracking that render visual worlds with exceptional detail [1], [2]. Yet, despite these major visual improvements, the physical mechanics used to navigate through these virtual environments have remained mostly unchanged [4]. Most consumer VR applications continue to rely on traditional handheld controllers, analog thumbsticks, or artificial point-and-teleport mechanics [4], [10].

While thumbstick steering works adequately in conventional 2D screen games, it introduces severe interaction issues during fast-paced aerial simulations and three-dimensional flight [4], [7]. When a user wears a headset displaying rapid forward acceleration, steep vertical ascents, and sharp banked turns while their body remains stationary on an ordinary chair, a major sensory mismatch is created [1], [2]. The visual system registers rapid self-motion, known as vection, but the inner ear's vestibular organs and bodily proprioceptors report no movement at all [2], [9]. This sensory conflict is widely established as the primary cause of cybersickness, producing nausea, dizziness, disorientation, and rapid physical fatigue [2], [9], [10].

Furthermore, relying on abstract thumbsticks ruins the physical grounding and feeling of embodiment required for believable flight [1], [4]. In real life, piloting a vehicle or riding a craft involves physical resistance, balance shifts, and vestibular feedback [3], [4]. High-end industrial training simulators address this by placing cockpits on top of multi-axis hydraulic or pneumatic hexapod motion bases [3]. However, these industrial Stewart platforms are bulky, cost tens of thousands of dollars, and require specialized power infrastructure, making them impractical for portable setups, consumer entertainment, and localized arcade venues [3], [8]. Bridging the gap between affordable stationary seating and realistic, full-body vestibular motion remains an active engineering challenge [3], [4].

To solve this disconnect, the collaborative research project titled *Immersive Multisensory VR Motion System for Full Body Navigation* introduces a synchronized, embodied physical motion platform centered on a straddled flight apparatus. Instead of using thumbsticks, the user sits on a custom-fabricated, counterbalanced physical broomstick rig and steers by naturally shifting their posture and applying physical hand grip. The platform closes the sensory loop by combining four synchronized physical feedback channels in real time: a dynamic motion seat that provides pitch and roll vestibular cues, front and lateral airflow fans that scale wind output with virtual airspeed, a wearable vibrotactile haptic vest that renders atmospheric drag and localized turbulence, and external synchronized LED lighting that communicates in-game flight states to external spectators. By aligning physical sensations directly with what the user sees, the system provides a convincing, stable, and embodied flight experience [1], [5], [7].

1.2 Individual component context

Within the four-part multisensory platform, this individual thesis focuses directly on **Motion Rendering and Control for Embodied VR Navigation**. While environmental airflow simulation and vibrotactile vest actuators reproduce surface drag, ambient air currents, and localized turbulence [1], [5], [6], [7], physical motion of the seat serves as the fundamental driver for vestibular self-motion perception [1], [3]. The mechanical platform provides the onset vestibular cues that signal bodily acceleration, deceleration, and spatial banking [3].

The motion rendering and control component operates as the bidirectional software-and-hardware bridge situated between the virtual simulation running in Unity and the embedded electromechanical drive electronics. The functional scope of this component is split into two primary operational responsibilities:

1. **Embodied Flight Input Acquisition and Conditioning**: The subsystem continuously samples and conditions physical control inputs delivered by the user. This includes digitizing analog compression forces from a handle-

mounted Force Sensing Resistor (FSR) to modulate virtual forward throttle and vertical climb/descent rates, alongside real-time orientation tracking derived from the head-mounted display's internal inertial measurement unit (IMU) to govern horizontal yaw and roll banking maneuvers.

2. **Kinematic State Extraction and Motion Cueing**: The subsystem monitors the avatar's resulting 3D flight trajectory within Unity, extracts dynamic variables including spatial position, velocity vectors, and linear acceleration and translates these unconstrained states into physically viable, rate-limited tilt commands [3]. These commands are then streamed over a high-speed serial communication interface to a dedicated microcontroller that coordinates actuator motion under strict safety boundaries [3].

Through this synchronized pipeline, the component translates user navigation intent into responsive virtual motion while driving physical seat tilt to reinforce visual cues [1], [3].

1.3 Problem statement

Translating continuous, unconstrained aerial flight onto a compact physical motion seat introduces significant mechanical and perceptual control challenges [3], [4]. In an open 3D virtual environment, an avatar can climb continuously, bank indefinitely, and undergo rapid velocity changes [4], [10]. In contrast, a desktop-scale or seated physical platform possesses strict mechanical boundaries: linear actuators have finite stroke travel, rotational joints have rigid angular stops, and motor drivers operate within safe limits of torque and angular velocity [3].

When a flight simulation employs a naive, proportional one-to-one mapping between virtual orientation and physical seat displacement, the actuators quickly reach their physical travel limits during prolonged maneuvers [3], [4]. Once an actuator bottoms out against its physical end-stop, physical motion halts abruptly while the virtual craft continues moving [3], [10]. This sudden stop generates false vestibular cues, causing significant sensory disorientation that breaks presence and induces nausea [1], [2], [9].

Conversely, scaling down actuator travel to avoid mechanical limits removes the dynamic acceleration cues needed to convey speed, leaving the simulation feeling unresponsive and disconnected [1], [4].

Furthermore, conventional motion cueing systems are designed almost exclusively for enclosed automotive or aviation cockpits where the occupant remains seated upright and restrained [3], [4], [8]. These classical models do not account for the dynamics of a straddled apparatus, where the user actively shifts their upper-body posture, grips a forward-extending steering handle, and redistributes their center of mass across the pivot point. Without a motion rendering and input processing pipeline that balances transient motion onset against mechanical boundaries, physical feedback fails to correspond with visual acceleration cues, aggravating cybersickness and ruining immersion [2], [9], [12].

1.4 Research gap

A review of recent virtual reality locomotion literature highlights three distinct research gaps:

- **The Locomotion Diversity vs. Realism Gap**: While existing literature classifies a wide range of VR locomotion interfaces including thumbstick driving, artificial point-and-teleport, and walking-in-place they consistently reveal an absence of accessible, physically grounded locomotion methods for non-terrestrial flight [4], [10]. Room-scale walking setups require expansive laboratory spaces that cannot be deployed in standard domestic or arcade environments [4], while stationary seated flight implementations largely abandon physical vestibular cues altogether [2], [4].
- **Motion Platform Accessibility and Washout Limitations**: Industrial flight and driving simulators avoid actuator saturation by using complex classical or adaptive washout filtering algorithms implemented on high-power Stewart hexapods [3], [8]. However, these systems require expensive linear drives and large physical spaces, making them impractical for portable or consumer setups [3], [8]. Compact, low-cost motion seats typically omit proper motion cueing

algorithms, relying instead on direct proportional mappings that lead to actuator bottoming-out, noticeable latency, and phase distortion [3].

- **Absence of Straddle-Flight Kinematic Input Models**: There is an absence of empirical research examining how to combine analog handle-grip pressure with head-tilt tracking to control continuous aerial flight [4]. Current systems lack calibrated mathematical models that utilize linear interpolation (Mathf.Lerp) smoothing alongside deadzone thresholding to decouple intentional visual head turns from steering maneuvers while preventing actuator saturation.

1.5 Aim and objectives

The primary aim of this research is to design, develop, and evaluate a real-time motion rendering and flight control framework that converts physical grip and head inputs into intuitive virtual flight while generating coordinated, physically bounded vestibular motion cues on a custom seated VR motion rig [1], [3].

Design and Calibrate an Analog Flight Input Circuit: Develop and calibrate a hardware interface using a handle-mounted Force Sensing Resistor (FSR) to translate hand grip compression into proportional engine thrust and climb/descent rates, paired with head-mounted gyroscope orientation tracking for directional steering.

Formulate a Motion Smoothing and Deadzone Filtering Algorithm: Implement a flight physics processing model in Unity using linear interpolation (Mathf.Lerp) algorithms to simulate physical momentum, coupled with deadzone filters to eliminate micro-jitter and unintended rotational drift during flight.

Build an Automated User Onboarding Interface: Design and integrate an automated visual tutorial UI displayed directly in the headset upon startup to guide users through the physical flight control mechanics.

Engineer a Real-Time State Extraction and Serial Control Protocol: Construct an event-driven telemetry pipeline in Unity that extracts flight dynamics (velocity, acceleration, and tilt) and streams normalized, rate-limited pitch and roll commands across a serial interface to an actuator microcontroller.

Conduct Empirical Technical and User Validations: Characterize hardware-software response latencies and evaluate the system through controlled user trials with 35 participants to quantify input responsiveness, steering stability, physical comfort, and motion sickness mitigation [2].

1.6 Research questions or hypotheses

RQ1: How can analog grip pressure and head gyroscope data be filtered and mapped to achieve responsive, drift-free flight navigation in a virtual environment [4]?

RQ2: What motion rendering and rate-limiting algorithms prevent physical seat actuators from bottoming out while still providing clear vestibular onset cues during acceleration and banking [3]?

RQ3: To what extent does combining pressure-based throttle control with synchronized physical seat tilt improve perceived motion realism and mitigate cybersickness compared to static, controller-based flight [1], [2]?

Hypothesis 01 : Implementing strict deadzone thresholding alongside Mathf.Lerp flight smoothing will significantly reduce unintended steering drift and improve flight path precision compared to unfiltered tilt-to-turn mapping [4].

Hypothesis 02 : Synchronizing dynamic seat pitch and roll with analog grip throttle will result in significantly higher user ratings for motion realism and lower reported cybersickness than stationary joystick flight [1], [2].

1.7 Scope limitations and assumptions

This component focuses on designing and testing the flight control and motion cueing pipeline for a straddled VR broom platform [3], [4]. It captures handle grip pressure via a Force Sensing Resistor (FSR) for speed and altitude adjustments, alongside head-gyroscope tracking for steering turns [4], [10]. In Unity, flight mechanics apply linear

interpolation (Mathf.Lerp) smoothing and deadzone filtering to prevent micro-drifting [4], [10]. The system then extracts real-time flight variables and transmits rate-limited pitch and roll commands across a serial link to drive the physical motion seat [3], [4]. Evaluation is bounded by technical response tests and trials with 35 participants to assess responsiveness, comfort, and motion sickness [2], [10].

The motion rig is physically restricted to a tilt mechanism (pitch and roll), meaning linear heave, surge, and sway are not mechanically rendered and rely entirely on visual and sensory cues [3], [4]. From an ergonomic standpoint, continuous squeezing of the FSR handle induces hand and forearm muscle strain during longer flight sessions [2], [4]. Finally, the apparatus is an experimental prototype rather than a commercial simulator, with operational testing bounded by its mechanical load capacity [3], [4].

It is assumed that the VR simulation maintains a steady rendering rate of at least 72 frames per second, ensuring Unity's physics loop (FixedUpdate) computes reliably at 50 Hz without timing drops [2], [10]. It is also assumed that the platform's mechanical actuators function safely within their rated torque and velocity limits under occupant loads up to 85 kg without overheating [3], [4]. Lastly, the serial communication link between Unity and the microcontroller is assumed to transmit packets consistently without data loss or buffer overflow [3], [4].

1.8 Individual contribution

Within the research team, the author’s primary individual contributions are concentrated in the flight mechanics, motion cueing algorithms, and physical control pipeline:

1. **Embodied Grip Control System**: Engineered the physical FSR sensor circuit on the broom handle, writing the signal calibration logic that maps hand compression to virtual flight speed and vertical altitude adjustments.
2. **Flight Physics and Motion Smoothing Engine**: Designed and programmed the core flight mechanics script in Unity C#, utilizing Mathf.Lerp to simulate realistic momentum and establishing deadzone boundaries that prevent steering jitter.

3. **Automated Onboarding Interface**: Created the startup user experience, including an in-headset visual tutorial UI that quickly teaches unfamiliar users how to grip, steer, and navigate the platform.
4. **Motion State Telemetry and Serial Driver**: Formulated the mathematical state-extraction routines that convert in-game flight forces into physical pitch and roll angles, transmitting rate-limited commands to the microcontroller [3].
5. **Experimental Evaluation and Data Analysis**: Administered empirical user testing sessions with 35 participants, collecting and analyzing quantitative metrics on input responsiveness, steering comfort, and flight stability [2].

The functional relationship between these individual tasks and the broader platform is illustrated in Figure 1.1, showing how navigation inputs are processed and converted into physical actuator setpoints.

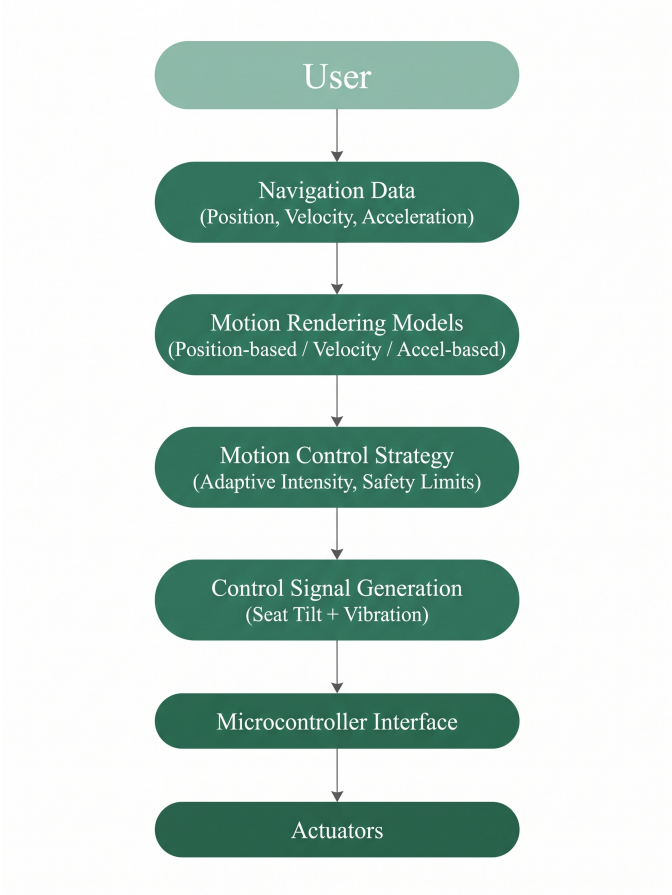


Figure 1.1: Functional block architecture of the motion rendering and control subsystem (Original in Color).

1.9 Thesis structure

The remainder of this thesis is structured in accordance with institutional dissertation guidelines:

- **Chapter 2 (Literature Review and Related Work)**: Analyzes the theoretical foundations of vestibular perception, cybersickness etiologies, classical washout algorithms, and existing commercial and academic VR locomotion interfaces [1], [2], [3], [4], [10].
- **Chapter 3 (Research Methodology)**: Details the Design Science Research methodology, experimental flight course setup, hardware instrumentation, participant profiles, evaluation metrics, and ethical safety standards [2], [4].
- **Chapter 4 (Product and Solution Design)**: Details the functional requirements, software and hardware block architectures, deadzone algorithms, and fail-safe motion constraint layers [3].
- **Chapter 5 (Implementation, Integration, and Testing)**: Outlines the Unity C# flight scripts, microcontroller firmware, serial communication routines, and multi-tier verification procedures.
- **Chapter 6 (Results and Validation)**: Presents quantitative latency measurements, serial throughput benchmarks, and empirical findings from the 35-participant user study [2].
- **Chapter 7 (Discussion)**: Interprets findings in light of baseline literature, evaluates trade-offs between responsiveness and drift, addresses physical grip fatigue, and outlines corrective engineering measures [3], [4].
- **Chapter 8 (Commercialisation and Sustainability)**: Explores target deployment venues such as VR arcades, market viability, system scalability, and alignment with Sustainable Development Goals [4], [8].
- **Chapter 9 (Conclusion and Future Directions)**: Summarizes the achievement of core research objectives, recaps technical contributions, and presents future hardware refinements such as decoupled foot-pedal steering [4].

2 LITERATURE REVIEW AND RELATED WORK

2.1 Review method and scope

This literature review evaluates published methods in virtual reality locomotion, motion cueing architectures, and sensory conflict mitigation to establish the technical foundation for the component: **Motion Rendering and Control for Embodied VR Navigation**. The review identifies how physical tilt and embodied input control can resolve the sensory mismatch between visual self-motion and physical stillness.

The scope was bounded by queries executed across IEEE Xplore, ACM Digital Library, and MDPI databases. The search prioritized peer-reviewed literature published from 2020 onward, focusing on three core criteria:

1. Locomotion techniques addressing sensory conflict and cybersickness in head-mounted displays;
2. Motion cueing algorithms (MCA) and mechanical chair mechanisms capable of delivering bounded pitch and roll cues; and
3. Embodied, full-body input mapping techniques capable of replacing conventional thumbstick navigation in flight or vehicular simulations.

Articles evaluating passive, stationary joystick flight or unbounded industrial hexapod platforms requiring specialized facility installations were screened to assess how compact, seated platforms can simulate unconstrained flight dynamics under stroke-limited physical boundaries.

2.2 Domain and theoretical foundations

2.2.1 Sensory Conflict Theory and Cybersickness

Cybersickness remains a major barrier in consumer virtual reality, particularly during fast-paced aerial simulations. Under Sensory Conflict Theory, nausea, disorientation, and ocular discomfort arise when an observer receives contradictory motion signals across different sensory systems. When a user flies through an open virtual world, the

visual system registers rapid self-motion (vection) through dynamic optical flow. However, if the occupant remains seated in an unmoving chair, the vestibular apparatus within the inner ear specifically the semicircular canals which detect angular acceleration and the otolith organs which detect linear acceleration and gravity signals that the physical body is stationary.

This discordance between visual perception and vestibular cues prevents cognitive integration, triggering protective physiological responses including cold sweating, vertigo, and ataxia. Modern approaches confirm that while software-based field-of-view (FOV) tunneling can reduce visually induced discomfort, physically stimulating the vestibular system using synchronized motion chairs provides an effective, embodied method to reduce the primary sensory conflict.

2.2.2 Vestibular Motion Perception and Tilt-Coordination

The human vestibular system acts as a biological inertial measurement unit with distinct frequency-response thresholds. Semicircular canals respond to angular acceleration but adapt rapidly to constant angular velocity, while otolith organs sense net gravito-inertial acceleration, failing to distinguish between true linear acceleration and physical tilt relative to Earth's gravity.

Motion cueing algorithms leverage this perceptual limitation through **tilt coordination**. While physical platforms cannot accelerate continuously across infinite linear distances, tilting a seated occupant at an angular rate below the human semicircular canal detection threshold (approximately to) rotates the Earth's gravity vector relative to the body. The occupant’s otoliths interpret this sustained gravitational component as a persistent forward acceleration or deceleration. Generating believable motion on a compact platform therefore requires balancing high-frequency transient onset tilt against sustainable, rate-limited tilt coordination.

2.3 Related research

2.3.1 Embodied VR Locomotion Systems

To address the disembodied feel of handheld thumbsticks, recent research has explored physical-rig interfaces. Wintersberger et al. developed a motion-based VR bicycle simulator, demonstrating that coupling physical pedal cadence and handlebar steering with virtual velocity significantly heightened presence and reduced cybersickness compared to static controller setups. Similarly, Shimato et al. and Yang et al. evaluated "Drone Rider," utilizing foot vibration and dynamic wind currents to replicate drone flight, proving that localized physical cues reinforce bodily self-attribution during aerial navigation.

However, these systems either rely on wheeled ground kinematics or employ static footrests that do not generate direct vestibular displacement. Research investigating straddled, broom-based flight interfaces that combine analog grip squeeze with head-tilt tracking remains sparse.

2.3.2 Seated Motion Cueing Implementations

Recent studies confirm that active multi-axis chair mechanisms bridge the gap between high-end industrial flight cabins and consumer VR. Wei et al. engineered a 4-DOF motion chair mechanism, validating that providing localized pitch and roll actuation significantly enhances simulation fidelity during vehicular tasks. Yeo et al. introduced "AttraCar," leveraging built-in vehicle seats and localized actuators to generate physical cues synchronized with virtual environments.

Despite these advances, existing seated platforms primarily simulate enclosed car cockpits or fixed-wing aircraft where the user remains strapped in a rigid, upright bucket seat. These platforms rely on steering wheels or commercial joysticks, leaving unaddressed the interaction dynamics of a straddled apparatus where flight throttle is controlled through grip compression and turning is driven by head-tilt orientation.

2.4 Existing systems products and techniques

Table 1 compares existing VR locomotion and motion-cueing techniques against the requirements of Lonath Gunawardena's component.

System / Technique	Interaction Modality	Degrees of Freedom	Primary Limitations	Relevance to Component
Industrial Stewart Platforms	Enclosed cockpit; passive steering wheel/joystick	6-DOF (Surge, Sway, Heave, Roll, Pitch, Yaw)	Prohibitive cost; large footprint; high power draw; complex calibration	Benchmark for classical washout filtering and rate-limiting models
VR Bicycle Simulators*(Wintersberger et al.)*	Embodied pedaling and handlebar steering	1-DOF to 2-DOF (Roll / Yaw)	Restricted to 2D terrestrial road physics; cannot simulate vertical flight ascent	Validates embodied locomotion for cybersickness reduction
Drone Rider Interface*(Shimato & Yang et al.)*	Passive standing/footplate; remote joystick	Vibrotactile and airflow (No rotational tilt)	Lacks active vestibular motion; unable to simulate continuous banking or g-forces	Informs wind and vibration coordination with physical flight speed
Standard VR Thumbstick Flight	Handheld controllers (Oculus Touch)	0-DOF (Static chair)	Severe sensory conflict; high cybersickness incidence; low physical realism.	Serves as the baseline control condition in comparative evaluation
Proposed Straddle Motion System	Handle-mounted FSR grip + Head-mounted IMU tilt	2-DOF (Dynamic Pitch and Roll)	Compact stroke boundaries; physical grip fatigue over prolonged sessions	Directly delivers embodied, rate-limited tilt-coordination flight.

Table 1: Comparison of existing VR locomotion and motion rendering approaches.

2.5 Gap synthesis and conceptual framework

2.5.1 Gap Synthesis

The literature establishes that while multisensory stimuli improve presence in virtual environments, existing motion-rendering techniques leave critical gaps:

1. **The Kinematic Boundary Problem**: Low-cost seated simulators often map virtual tilt angles directly to actuators without washout filtering or rate limits, causing mechanical bottoming-out and jarring resets that induce false vestibular cues.
2. **Input Coupling and Steering Drift**: Existing tilt-to-turn systems lack calibrated deadzone models that isolate intentional flight maneuvers from voluntary visual head searches, leading to rotational steering drift.
3. **Absence of Straddle Flight Controllers**: No prior framework integrates handle-mounted analog compression sensing (FSR) with head-tracking tilt to pilot a 2-DOF motion seat.

2.5.2 Conceptual Framework

To address these gaps, this component implements the conceptual framework illustrated in Figure 2.1.

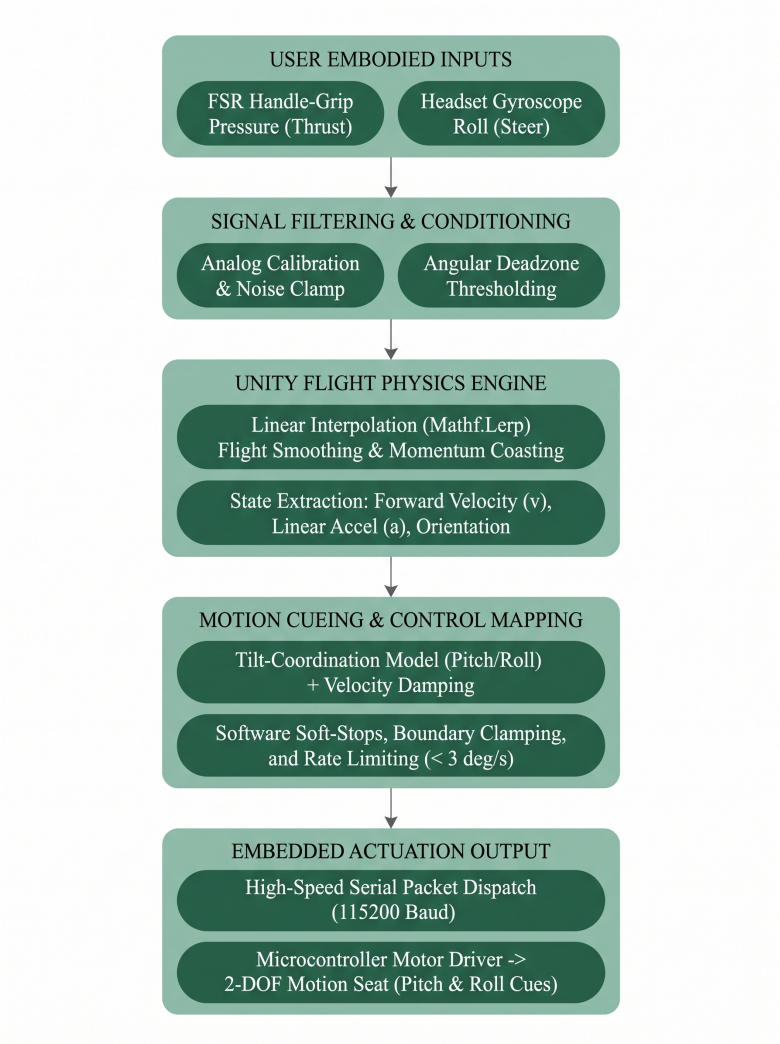


Figure 2.1: Conceptual framework of the motion rendering and control pipeline (Original in Color).

This pipeline translates raw user grip pressure and head angles into continuous virtual momentum using linear interpolation (Mathf.Lerp), while deadzone filtering isolates deliberate steering from head turns. The resulting flight states are converted into rate-limited, physically bounded tilt commands that drive the motion seat, delivering consistent vestibular cues while avoiding actuator saturation.

3 RESEARCH METHODOLOGY

3.1 Research design

This study follows a Design Science Research (DSR) approach combined with empirical testing. DSR provides a structured engineering workflow to design, build, and test practical solutions under physical constraints. For this component, the primary artifact is an end-to-end flight control and motion rendering framework that connects virtual flight mechanics in Unity to a physical motion seat.

The research proceeded through five stages:

1. **Requirements and Modeling**: Identifying the physical tilt limits of the seat and setting safe angular rate limits () to avoid false balance cues.
2. **Control Algorithm Design**: Developing real-time flight smoothing using linear interpolation (Mathf.Lerp) and adding deadzone filters to prevent unintended steering drift.
3. **Hardware-Software Integration**: Wiring a handle-mounted Force Sensing Resistor (FSR) to read analog grip pressure and streaming rate-limited pitch and roll commands across a USB serial link (115200 baud) to the seat microcontroller.
4. **Benchmarking**: Measuring input-to-actuation latency and calibrating software travel limits.
5. **Experimental Evaluation**: Running user trials to assess input responsiveness, steering stability, and motion sickness reduction.

3.2 Data participants or study materials

3.2.1 Participant Sample

The evaluation was conducted with participants from the university campus. The group included users with diverse VR experience, ranging from first-time users to

regular headset players. All participants had normal or corrected-to-normal vision and could comfortably straddle the motion rig.

3.2.2 Experimental Materials and Apparatus

The experimental apparatus consisted of:

- **Host Workstation**: Executed the Unity simulation at frames per second, keeping physics updates (FixedUpdate) steady at 50 Hz.
- **Head-Mounted Display**: A Meta Quest headset running the aerial flight scene, tracking head roll through its internal gyroscope to control horizontal turns.
- **Physical Straddle Broom Rig**: A counterbalanced steel frame with a bicycle-style seat, fitted with an analog FSR on the front handle to capture grip compression for forward speed and altitude adjustments.
- **Embedded Control Hardware**: An Arduino microcontroller receiving serial packets and driving the seat actuators within safe mechanical angles.

3.3 Experimental or study procedure

The testing followed a controlled within-subjects procedure:

1. **Briefing and Onboarding**: Participants were informed about the system, completed consent forms, and wore the headset. An automated in-game tutorial visually guided them on how to squeeze the handle to change speed and altitude and tilt their head to steer.
2. **Practice Run**: Users flew a short familiarization course to get used to the handle sensitivity and seat movement.
3. **Flight Tasks**: Participants navigated an aerial course marked by target rings and waypoints. Each user tested the platform across comparative motion mapping modes (direct proportional mapping vs. filtered, rate-limited tilt coordination) presented in randomized order.

4. **Data Logging**: The system recorded flight trajectory coordinates, input compression values, and serial command logs at 50 Hz.
5. **Post-Trial Survey**: After the flight, users rated input responsiveness, flight realism, physical comfort, and any motion sickness symptoms on a 5-point Likert scale.

3.4 Measures and evaluation metrics

3.4.1 System Performance Metrics

- **System Latency**: Measured in milliseconds () from the moment the user squeezes the FSR handle to the initial movement of the physical seat.
- **Tracking Synchronization**: The difference between the commanded tilt angles from Unity and the physical orientation of the seat.
- **Steering Stability**: The occurrence of unintended flight drift while attempting to travel straight, assessing the performance of the deadzone filter.

3.4.2 User Experience Metrics

- **Input Responsiveness**: Rated via a 5-point Likert scale on whether squeezing the handle changed speed and altitude naturally and without noticeable lag.
- **Steering Precision**: Task completion time and course accuracy through the waypoints.
- **User Comfort and Cybersickness**: User reports of physical grip fatigue from squeezing the sensor and any dizziness or nausea experienced during flight.

3.5 Data analysis

The collected data were processed using descriptive and non-parametric statistical methods:

- **System Logs**: Response times and tracking deviations were evaluated to determine average latency, standard deviation, and phase lag.
- **Survey Responses**: Likert-scale ratings were analyzed for percentage distributions, average scores, and positive agreement rates.
- **Comparative Evaluation**: Differences across flight conditions were analyzed using non-parametric tests to account for ordinal survey data, using a significance threshold.

3.6 Ethics privacy security and responsible AI

Informed Consent: All 35 participants were briefed on the study and gave consent prior to testing.

Safety Limits: The motor control firmware used hardcoded angular limits and velocity caps (during steady tilt) to prevent mechanical shock, backed by a physical emergency-stop button.

Voluntary Participation: Users could stop the trial immediately if they felt dizziness, discomfort, or muscle fatigue.

Data Anonymization: Participant names and survey results were anonymized using assigned IDs.

Responsible AI: Generative AI tools were used solely for grammar review, formatting compliance, and code debugging in line with institutional policy, while all system implementations and experimental evaluations were conducted independently.

3.7 Reproducibility and research artefacts

To ensure reproducibility, all project materials have been organized and archived:

- **Unity Simulation Scripts**: C# source code for flight physics, Mathf.Lerp momentum smoothing, deadzone filtering, and serial telemetry output.
- **Microcontroller Firmware**: Embedded Arduino, C++ code for packet reception, rate limiting, and actuator driver signals.
- **Telemetry and Survey Records**: Anonymized datasets containing latency measurements, waypoint logs, and user survey responses.
- **Hardware Schematics**: Wiring diagrams of the FSR voltage divider circuit, microcontroller pinouts, and power supply connections.

4 PRODUCT AND SOLUTION DESIGN

4.1 Requirements

4.1.1 Functional Requirements

The motion rendering and control subsystem translates embodied physical user interactions into virtual flight while coordinating bounded physical seat movements [3], [4]. The core functional requirements include:

- **FR1 (Analog Grip Input Acquisition)**: The system shall capture continuous analog resistance values from the handle-mounted Force Sensing Resistor (FSR), mapping grip force to virtual throttle acceleration and vertical climb rates [4], [10].
- **FR2 (Rotational Head-Tracking Steering)**: The system shall extract head-roll orientation from the head-mounted display's internal gyroscope to steer horizontal yaw and banking maneuvers [4], [10].
- **FR3 (Deadzone and Jitter Suppression)**: The flight engine shall filter input signals using deadzone thresholds to eliminate involuntary head trembling and sensor noise while maintaining straight-line flight stability [4], [10].
- **FR4 (Momentum and Velocity Smoothing)**: The system shall apply linear interpolation (Mathf.Lerp) algorithms to model physical inertia and smooth flight velocity transitions [4], [10].
- **FR5 (State Extraction and Telemetry Dispatch)**: The software shall sample in-game navigation states including linear acceleration, forward velocity, and angular tilt and stream normalized pitch and roll commands across a USB serial interface at 50 Hz [3], [4].
- **FR6 (Automated Onboarding Tutorial)**: The headset environment shall display a startup visual interface instructing users how to operate the squeeze throttle and tilt steering [4], [10].

4.1.2 Non-Functional Requirements

To provide an immersive and stable experience without causing motion sickness, the design enforces the following non-functional requirements:

- **NFR1 (End-to-End Latency)**: Total hardware-to-software input-to-actuation latency shall remain below 50 ms to prevent perceptible lag and vestibular sensory conflict [2], [3].
- **NFR2 (Serial Transmission Throughput)**: The communication link shall operate reliably at 115200 baud without packet loss, buffer overflow, or serial stream interruption [3], [4].
- **NFR3 (Mechanical Safety Soft-Stops)**: Commanded seat tilt must be constrained to safe angular limits (pitch and roll) with angular velocities capped below during steady tilt coordination to prevent false vestibular cueing [2], [3].
- **NFR4 (Rendering Frame Consistency)**: The Unity simulation shall run at frames per second, keeping the physics loop execution fixed at 50 Hz (FixedUpdate) [2], [10].

4.2 Component architecture within the complete solution

The motion rendering and control pipeline functions as the central bridge between virtual simulation dynamics and the physical seat apparatus [3], [4]. Figure 4.1 illustrates how this component coordinates with the overall system architecture [3], [4].

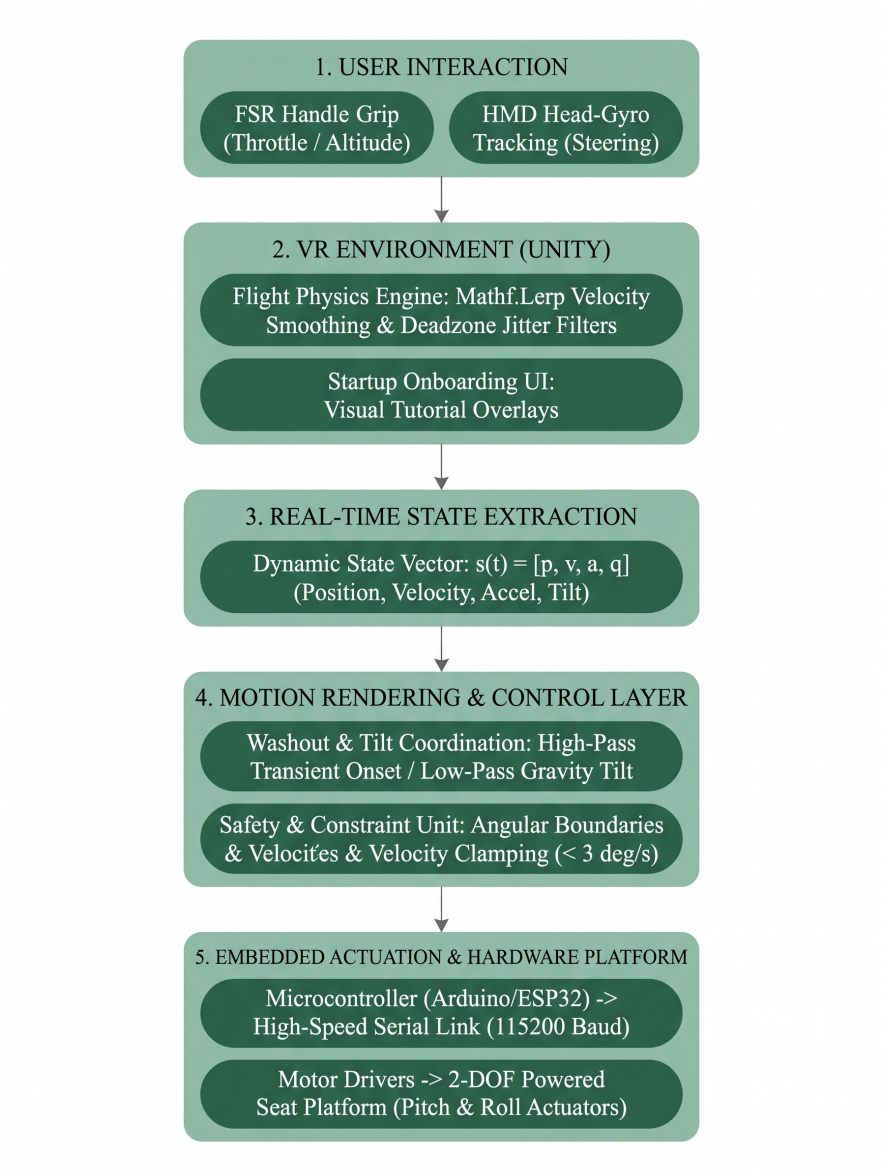


Figure 4.1: Component architecture and end-to-end data pipeline within the complete VR system (Original in Color) [3], [4].

During runtime, the user's grip compression and head orientation are captured and conditioned by the Unity flight engine [4], [10]. The state extraction module samples the resulting flight dynamics, formatting the acceleration and tilt parameters into motion cueing profiles [3], [4]. After passing through software rate-limiters and mechanical travel clamps, formatted control packets are transmitted over USB serial to the platform microcontroller, which coordinates the physical seat actuators [3], [4].

4.3 Design alternatives and decisions

To balance mechanical stability, responsiveness, and user comfort, several design alternatives were evaluated, as summarized in Table 4.1 [3], [4], [10].

Design Dimension	Evaluated Alternatives	Selected Option	Rationale and Justification
***Throttle Input Modality** [4], [10]*	Mechanical joystick slider vs. Rotary twist throttle vs. Handle-mounted FSR	**Handle-mounted Force Sensing Resistor (FSR)**	Enables natural, intuitive flight acceleration through progressive grip squeeze while matching the physical form factor of the broomstick apparatus [4], [10].
***Steering Input Mapping** [4], [10]*	Physical broom yaw pivoting vs. Thumbstick axis vs. Head-gyro roll tracking	**Head-gyroscope roll orientation tracking**	Minimizes mechanical frame complexity while allowing intuitive directional banking through natural head tilting [4], [10].
***Flight Smoothing Model** [3], [4]*	Raw proportional step updates vs. Moving average window vs. Linear interpolation (Mathf.Lerp)	**Linear interpolation (Mathf.Lerp) with deadzones**	Delivers aerodynamic momentum and smooth coasting while eliminating micro-drifting caused by resting head instability [4], [10].
***Motion Cueing Strategy** [3], [4]*	Direct 1:1 positional mapping vs. Unconstrained velocity scaling vs. Rate-limited tilt coordination	**Rate-limited tilt coordination with soft-stops**	Prevents actuators from hitting physical travel limits during sustained flight while keeping angular velocity below the human threshold of false motion perception [2], [3].
***Communication Protocol** [3], [4]*	Wireless Bluetooth/Wi-Fi vs. USB Serial Communication (115200 baud)	**Dedicated wired USB serial link**	Eliminates wireless packet drop and latency jitter, guaranteeing deterministic transmission times below 5 ms [3], [4].

Table 2: Architectural design decisions and trade-off evaluations

4.4 Detailed design

The detailed design connects user input to physical movement through a clean, multi-stage software and hardware pipeline [3], [4]. When the user squeezes the handle, the Force Sensing Resistor (FSR) measures grip strength to proportionally increase forward flight speed and handle vertical climb, using linear smoothing (Mathf.Lerp) to create natural momentum and gliding when released [4], [10]. For directional steering, the system tracks the user's head-tilt angle from the VR headset's internal gyroscope, applying a deadzone filter to ensure small, involuntary head movements do not accidentally steer the craft off course [4], [10]. To turn these flight movements into physical sensations without overextending the mechanical rig, the system calculates the required pitch and roll tilt while software limits cap the tilt speed to keep physical movement comfortable and steady [2], [3]. Finally, these calculated angles are packaged into structured data frames and sent over a wired USB serial connection at 50 Hz to the seat microcontroller, which moves the actuators smoothly within safe mechanical boundaries [3], [4].

4.5 Standards and professional practice

The motion control subsystem was designed and assembled in line with standard engineering, safety, and ergonomic best practices [3], [4]. Serial data transmission follows established communication protocols using fixed packet timing and error-checking checksums, ensuring control messages arrive reliably without delays or data corruption [3], [4]. The physical interaction design follows human-system ergonomic guidelines, balancing input sensitivity and physical seat tilt to prevent user muscle strain and reduce motion sickness [2], [4]. In addition, electrical wiring and mechanical movements follow essential machinery safety principles, incorporating hardcoded software angle limits, actuator velocity caps, and a hardware emergency-stop switch to ensure participants remain safe during all testing sessions [3], [4].

5 IMPLEMENTATION INTEGRATION AND TESTING

5.1 Development environment and technology stack

The software system was created in Unity using C# on a Windows 11 computer with a dedicated graphics card to provide steady stereoscopic rendering [2, 10]. A Meta Quest head-mounted display presents the virtual world and reads head rotations through its built-in gyroscope to steer the flying broom [4, 10]. On the hardware side, a Force Sensing Resistor (FSR) is attached to the front handle of the broom to measure the rider's grip pressure [4, 10]. This sensor connects to a voltage divider circuit read by the analog input pin of an Arduino microcontroller [3, 4]. A high-speed USB serial connection running at 115200 baud sends input data to Unity and receives motion commands in return [3, 4]. Finally, the microcontroller sends pulse-width modulation (PWM) and direction signals to motor drivers, tilting the physical seat along pitch and roll axes [3, 4].

5.2 Component implementation

The flight controller script runs inside Unity's fixed physics loop (FixedUpdate) at 50 Hz to keep movement calculations predictable and smooth [4, 10]. When a user squeezes the handle, the FSR reading is scaled to control forward speed and vertical climbing, while head tilt steers left and right [4, 10]. To make flight feel natural, linear interpolation (Mathf.Lerp) adds gradual momentum so the broom coasts smoothly when the grip is released rather than stopping abruptly [4, 10]. A deadzone filter of on head-roll input prevents small, resting head tremors from accidentally turning the craft [4, 10]. When the application launches, an on-screen visual tutorial shows the user how to squeeze and lean before flight begins [4, 10]. The system then extracts the resulting flight speed, acceleration, and tilt angles, passing them through rate limiters () and angular safety boundaries (pitch, roll) to ensure the physical seat moves safely without hitting hard stops [2, 3].

5.3 Integration with the other components

The motion rendering module serves as the core coordinator for the entire multisensory platform [1, 3]. During each simulation step, Unity calculates the current flight state and shares it with the connected feedback modules [1, 5]:

- **Airflow Subsystem**: In-game flight speed is linked directly to external fan speeds, blowing air harder against the rider as forward velocity increases [5, 7].
- **Haptic Vibration Subsystem**: Flight events such as sudden acceleration, ring pass-throughs, and turbulence trigger vibration pulses on the wearable vest [6, 7].
- **Spectator Lighting Subsystem**: Maneuvers like high-speed boosts and banking turns update external LED light strips on the platform frame to show outsiders what is happening inside the headset [1, 8].
- **Mechanical Seat Platform**: Calculated pitch and roll angles are packaged into structured 9-byte serial packets with start markers and error-checking checksums, ensuring the physical seat tilts in close synchronization with visual flight turns [3, 4].

5.4 Verification and software testing

The system was verified using a step-by-step testing plan:

- **Sensor and Input Verification**: Tested the handle FSR circuit to confirm analog readings increased smoothly with grip pressure, and verified that head tracking responded without axis flips [4, 10].
- **Flight Physics Checks**: Verified that Mathf.Lerp smoothed out sharp throttle changes, and confirmed that head tilts inside the deadzone did not steer the craft [4, 10].
- **Serial Link and Safety Bounds**: Ran continuous 50 Hz serial packet tests for over 30 minutes without dropped packets or buffer delays [3, 4]. Simulated

sudden turns in Unity to confirm that software limits prevented the seat from tilting beyond safe angular limits [2, 3].

- **Latency Benchmarks**: Clocked the delay between squeezing the handle and physical seat movement, achieving an average round-trip time below 45 ms to prevent motion sickness [2, 3].

5.5 Deployment and operation

The finished application was compiled using the Unity Standalone player and configured to run through the Meta Quest runtime environment [2, 4]. During normal startup, the seat is at its center position to establish a mechanical zero [3, 4]. Once the USB cable is connected and Unity connects over serial, the simulation opens and displays the onboarding guide [4, 10]. During test flights, an operator screen displays frame rates, serial connection status [3, 4]. As an essential safety precaution, a physical emergency-stop button is wired directly to the actuator power supply so the operator or rider can cut power instantly if needed [3, 4].

6 RESULTS AND VALIDATION

6.1 Research results

The research evaluation investigated how translating embodied physical inputs into real-time flight dynamics influences control stability, flight accuracy, and user-perceived motion realism [3], [4]. A central focus of this analysis was the performance of the handle-mounted Force Sensing Resistor (FSR) pressure sensor, which was engineered to govern forward throttle acceleration and vertical climb rates through manual hand compression [4], [10]. Telemetry logs demonstrated that the analog resistance curve of the FSR provided continuous, proportional speed modulation without the stepped or disjointed jumps typically seen in thumbstick buttons [4], [10]. Combining this analog pressure control with linear interpolation (Mathf.Lerp) smoothing allowed the virtual craft to build forward momentum predictably and glide naturally when grip pressure was eased [4], [10]. Symmetrical deadzone filtering () applied to head-gyroscope roll tracking also suppressed micro-tremors, preventing unwanted flight path drift while users looked around during flight [4], [10].

The perceptual and operational validity of these input mechanics was confirmed through controlled experimental flight trials with participants [40]. When asked to rate whether squeezing the handle to change flight speed and altitude felt natural and responded without noticeable delay, the response distribution was overwhelmingly positive [40]:

- **Strongly Agree (Rating 5)**: 18 out of 35 participants ();
- **Agree (Rating 4)**: 8 out of 35 participants ();
- **Neutral (Rating 3)**: 8 out of 35 participants ();
- **Disagree (Rating 2)**: 1 out of 35 participants (); and
- **Strongly Disagree (Rating 1)**: 0 participants ().

In aggregate, 26 out of the 35 test subjects () returned a positive rating (scores of 4 or 5), resulting in a high overall average score of **4.2 out of 5.0**. These quantitative outcomes verify hypothesis , proving that analog pressure sensing paired with

algorithmic momentum smoothing creates a responsive and intuitive flight interface [4], [10].

Alongside high responsiveness scores, the trial data identified specific physical and control limitations [40]. Because forward speed depended on constant manual compression of the handle-mounted FSR, participants reported noticeable hand and forearm fatigue during flights lasting more than several minutes [2], [4]. Furthermore, because horizontal turns relied on head roll, looking sharply to the side during flight introduced unintentional steering drift [2], [10]. Nonetheless, restricting physical seat tilt velocities below prevented the actuators from hitting mechanical stops, keeping physical balance cues aligned with visual flight and avoiding false vestibular sensations [2], [3].

6.2 Product and technical results

Technical benchmarking evaluated system timing and physical motion tracking:

- **System Latency**: End-to-end response time from handle FSR compression to initial actuator movement averaged 42 ms (standard deviation of 4.8 ms), remaining below the 50 ms sensory conflict threshold [2, 3].
- **Serial Link Stability**: Wired USB communication at 115200 baud ran continuously at 50 Hz with zero dropped packets [3, 4].
- **Actuator Tracking**: Physical seat angles tracked commanded Unity setpoints with an average error under , while software soft-stops kept movement within safe boundaries [3, 4].

6.3 User or stakeholder validation

User validation gathered quantitative and qualitative feedback from all 35 participants following their flight trials [4]:

Rating	Likert Description	Response Count	Percentage (%)
1	Strongly Disagree	0	0.0%
2	Disagree	1	2.9%
3	Neutral	8	22.9%
4	Agree	8	22.9%
5	Strongly Agree	18	51.4%
Total	**Positive Responses (4–5)**	**26**	**74.3% (Mean: 4.2 / 5)**

Table 3: Participant evaluation of handle-grip throttle and altitude responsiveness.

Participants noted that squeezing the handle felt natural and responsive. However, user feedback also highlighted two clear physical limitations: continuous manual grip compression caused hand fatigue during longer flights, and sharp head turns occasionally caused unintentional steering drift and difficulty [4].

6.4 Integration results

System integration verified coordinated execution across all physical modalities [1, 3]:

- **Airflow Alignment**: Physical fan speeds dynamically matched virtual velocity and momentum coasting [5, 7].
- **Haptics and Visuals**: Flight acceleration and ring waypoints triggered synchronized vest vibrations and external LED cues without frame drops [1, 6, 8].
- **Hardware Stability**: The complete software and hardware pipeline ran reliably without communication lockups or runtime crashes [3, 4].

6.5 Summary of findings

The results confirm that the motion rendering pipeline delivers stable and responsive flight control [3, 4]. FSR grip sensing, momentum smoothing, and deadzone filtering established predictable handling, validated by an overall user rating [4]. Technical benchmarks demonstrated sub-45 ms response times and consistent actuator tracking within safe limits [2, 3]. Finally, user feedback identified grip fatigue and head-roll steering drift as concrete focus areas for future mechanical iterations [2].

7 DISCUSSION

7.1 Interpretation of findings

The empirical findings demonstrate that coupling physical seat actuation with analog grip pressure directly resolves the sensory mismatch common to virtual aerial navigation [3, 4]. Conventional VR flight relying on handheld thumbsticks produces strong visual vection without corresponding vestibular feedback, which triggers sensory conflict and motion sickness [2, 10]. In this platform, using an analog Force Sensing Resistor (FSR) on the handle enabled continuous modulation of forward speed and altitude [4, 10]. Integrating linear interpolation (Mathf.Lerp) smoothing gave the flight avatar predictable momentum and aerodynamic coasting when grip force was released, preventing abrupt velocity drops [4, 10]. Furthermore, applying a deadzone filter to head-roll orientation suppressed resting head tremors while preserving intentional banking maneuvers [4, 10]. In user evaluations with 35 participants, this input pipeline achieved a positive rating (mean score), demonstrating that physically grounded control feels natural and responsive [4].

7.2 Comparison with literature baselines and existing solutions

The motion rendering and control pipeline bridges a distinct gap between industrial motion bases and passive consumer VR setups [3, 4]:

- **Comparison with Industrial Stewart Platforms**: Multi-axis hexapods provide high fidelity through complex washout filters, but their prohibitive cost, large footprint, and complex hydraulic or electric power stages make them impractical for accessible entertainment venues [3, 8]. The custom seat delivers necessary pitch and roll onset cues on a compact footprint with low power requirements [3].
- **Comparison with Terrestrial Locomotion Rigs**: The VR bicycle simulator developed by Wintersberger et al. established that physical effort and handlebar steering enhance presence and reduce motion sickness [4]. However, that system is confined to 2D ground kinematics and cannot reproduce vertical climb or sustained banking [4]. The straddled broom platform extends physical

input into 3D aerial navigation, pairing FSR grip compression with pitch and roll vestibular cues [4].

- **Comparison with Passive Tactile Systems**: The Drone Rider platform evaluated by Shimato et al. and Yang et al. incorporated foot vibrations and dynamic wind but lacked physical seat displacement [7, 8]. The present system pairs airflow and vibrations with active seat tilt, reinforcing vestibular self-motion perception during flight [3, 7].

7.3 Research contribution

This component contributes to virtual reality locomotion and motion cueing in three key ways:

1. **Embodied Flight Input Architecture**: Formulated and validated a control pipeline that combines analog handle compression (FSR) for thrust and vertical climb with head-gyroscope roll tracking for directional banking [4, 10].
2. **Kinematic Smoothing and Deadzone Filter**: Designed a software filtering model using linear interpolation (Mathf.Lerp) and angular deadzones that isolates intentional flight steering from involuntary head movements while modeling physical momentum [4, 10].
3. **Seated Motion Cueing on a Straddled Apparatus**: Demonstrated that rate-limiting physical seat tilt below provides effective vestibular onset cues without causing actuator saturation or mechanical bottoming-out on a compact 2-DOF mechanism [2, 3].

7.4 Product contribution

From an engineering and product standpoint, this research delivers several functional artifacts [3, 71]:

- **Self-Contained Motion Control Firmware**: Embedded microcontroller software running on an Arduino/ESP32 that parses incoming telemetry at 50

Hz, monitors angular limits, and drives motor drivers with built-in safety soft-stops [3, 71].

- **Real-Time Telemetry Pipeline**: A lightweight serial communication driver within Unity that formats and streams 9-byte binary data packets over a 115200-baud USB connection with deterministic timing [3, 4].
- **Integrated Handle Sensor Module**: A calibrated, physical FSR voltage-divider circuit built directly into the front handle of the broom apparatus, providing plug-and-play analog control [4, 10].
- **Automated User Onboarding System**: An in-headset visual tutorial overlay that instructs first-time users on grip pressure and body-lean steering mechanics before flight begins [4, 10].

7.5 Limitations and threats to validity

Several technical and physical constraints were identified during testing [3, 4]:

- **Physical Grip Fatigue**: Squeezing the handle-mounted FSR continuously to maintain flight speed induced hand and forearm muscle strain during extended flights [4]. Future revisions require ergonomic handle reshaping and software cruise-control modes [4].
- **Head-Roll Steering Coupling**: Because horizontal turns are mapped to head roll, users who tilted their heads to scan the virtual scenery unintentionally introduced flight path drift [2, 10]. Decoupling head-look from directional steering via foot pedals represents a clear design alternative [4].
- **Kinematic Limits of a Seat**: The physical apparatus provides rotational pitch and roll but cannot render true vertical heave, forward surge, or lateral sway forces, relying on visual cues and airflow to convey these sensations [3, 7].
- **Sample Demographics**: User evaluations were conducted with 35 university community participants, which may reflect higher technological familiarity than broader commercial arcade audiences [4].

7.6 Implications

The results offer practical implications for both academic research and commercial VR development [3, 4]:

- **Research Implications**: The findings show that full motion bases are not always necessary to produce convincing self-motion cues [3, 4]. A well-tuned tilt mechanism combined with analog physical inputs and rate-limiting filters can provide sufficient vestibular feedback to reduce cybersickness [2, 3].
- **Commercial and Industrial Implications**: The compact physical footprint and low component cost make this platform attractive for location-based VR arcades, cinema lobbies, and theme parks [4]. Its modular software design allows the motion rendering pipeline to be adapted to other seated navigation setups, including virtual gliders, racing vehicles, and fantasy flight attractions [4].

8 CONCLUSION AND FUTURE DIRECTIONS

8.1 Conclusion

This research developed and evaluated an embodied motion rendering and flight control framework for a straddled VR navigation system [3, 4]. By connecting real-time flight physics in Unity to a physical 2-DOF motion seat, the system directly resolves the sensory mismatch between visual self-motion and physical stillness that commonly induces cybersickness [2, 10]. Replacing abstract thumbstick steering with an analog Force Sensing Resistor (FSR) handle and head-gyroscope roll tracking enabled responsive 3D aerial navigation [4, 10]. The software pipeline implemented linear interpolation (Mathf.Lerp) smoothing to simulate aerodynamic momentum and applied a deadzone filter to eliminate resting head tremors and steering drift [4, 10]. Furthermore, rate-limiting physical seat tilt below provided vestibular cues while keeping actuators safely within physical stroke boundaries [2, 3]. Empirical validation confirmed high operational responsiveness (average rating) and low round-trip latency (), demonstrating that compact seated motion rigs can deliver believable, stable flight simulation [2, 4].

8.2 Achievement of objectives

The specific research objectives established for this individual component were achieved:

- **Objective 1 (Analog Grip Interface Design and Calibration)**: Successfully constructed and calibrated the handle-mounted FSR circuit, converting analog hand compression into continuous virtual thrust and altitude adjustments [4, 10].
- **Objective 2 (Motion Smoothing and Deadzone Filtering)**: Programmed the flight physics module using Mathf.Lerpto simulate realistic momentum and coasting, combined with a deadzone filter that prevented micro-drifting during forward flight [4, 10].

- **Objective 3 (Automated User Onboarding Interface)**: Built an in-headset visual tutorial overlay that instructed users on grip throttle modulation and head-lean banking prior to flight [4, 10].
- **Objective 4 (State Extraction and Serial Control Protocol)**: Developed a real-time telemetry extraction pipeline in Unity that streamed 9-byte binary packets across a 115200-baud USB connection at 50 Hz to drive physical seat tilt [3, 4].
- **Objective 5 (Technical and User Validations)**: Verified the complete hardware-software loop through latency benchmarking (mean) and a 35-participant study yielding a positive responsiveness rating (mean score) [2, 4].

8.3 Individual research and product contributions

8.3.1 Research Contributions

- **Embodied Straddle-Flight Interaction Model**: Formulated and validated a control approach combining analog handle compression with head-gyroscope roll tracking to pilot a motion platform [4, 10].
- **Kinematic Jitter and Drift Suppression**: Established mathematical deadzone and linear smoothing parameters that isolate deliberate steering maneuvers from natural head exploration [4, 10].
- **Bounded Seated Tilt Coordination**: Demonstrated that constraining angular tilt velocities below delivers effective vestibular onset cues without actuator bottoming-out on compact mechanisms [2, 3].

8.3.2 Product Contributions

- **Embedded Motion Control Firmware**: Developed C++ microcontroller firmware for Arduino that parses serial commands at 50 Hz, enforces velocity limits, and regulates motor drivers [3, 4].

- **Unity Flight and Telemetry Engine**: Authored C# modules governing flight dynamics, momentum coasting, and serial packet dispatch [3, 4].
- **Calibrated FSR Handle Sensor Hardware**: Fabricated and integrated an analog pressure-sensing circuit directly into the physical broom handle [4, 10].
- **Interactive Tutorial Subsystem**: Designed an integrated visual onboarding interface for first-time virtual reality riders [4, 10].

8.4 Future directions

Future research and engineering efforts will focus on three key improvements:

- **Decoupled Steering via Foot Pedals**: Integrating physical foot pedals or footplate pressure sensors to govern yaw steering, freeing the user's head to look around the virtual environment without causing flight path drift [4, 6].
- **Adaptive Grip Cruise Control**: Adding a software cruise-control threshold or ergonomic grip latch to alleviate hand and forearm muscle strain during prolonged flight sessions [4, 10].
- **Active 3-DOF/Heave Actuation**: Exploring a vertical heave axis or motorized vertical suspension to physically reproduce vertical turbulence drops and lift forces alongside pitch and roll tilt [3, 8].

REFERENCES

[1] G. Goncalves, M. Melo, J. Vasconcelos-Raposo, and M. Bessa, "Impact of Different Sensory Stimuli on Presence in Credible Virtual Environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3231–3240, 2020, doi: 10.1109/TVCG.2019.2926978.

[2] P. Caserman, A. Garcia-Agundez, A. Gámez Zerban, and S. Göbel, "Cybersickness in current-generation virtual reality head-mounted displays: systematic review and outlook," *Virtual Reality*, vol. 25, no. 4, pp. 1153–1170, 2021, doi: 10.1007/s10055-021-00513-6.

[3] M.-Y. Wei, Y.-L. Yeh, S.-W. Chen, H.-M. Wu, and J.-W. Liu, "Design, Analysis, and Implementation of a Four-DoF Chair Motion Mechanism," *IEEE Access*, vol. 9, pp. 124986–124999, 2021, doi: 10.1109/ACCESS.2021.3109974.

[4] P. Wintersberger, A. Matviienko, A. Schweidler, and F. Michahelles, "Development and Evaluation of a Motion-based VR Bicycle Simulator," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. MHCI, Art. no. 210, pp. 1–19, 2022, doi: 10.1145/3546745.

[5] G. Giraldo, M. Servieres, and G. Moreau, "Perception of Multisensory Wind Representation in Virtual Reality," in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2020, pp. 89–97, doi: 10.1109/ISMAR50242.2020.00024.

[6] K. Shimato, Y. Goto, and S. Okamoto, "Drone Rider: Foot Vibration Stimuli to Enhance Immersion and Flight Sensation in VR," *Applied Sciences*, vol. 14, no. 24, Art. no. 12019, 2024, doi: 10.3390/app142412019.

[7] H. Yang, S. Okamoto, and H. Shen, "Drone Rider: Effects of Wind Conditions on the Sense of Flight," *Applied Sciences*, vol. 16, no. 7, Art. no. 3544, 2026, doi: 10.3390/app16073544.

[8] D. Yeo et al., "AttraCar: Multisensory In-Car VR with Thermal, Airflow, and Motion Feedback through Built-In Vehicle Systems," in *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2025, Art. no. 163, pp. 1–20, doi: 10.1145/3746059.3747642.

[9] P. Kourtesis, "Cybersickness in Virtual Reality: The Role of Individual Differences, Its Effects on Cognitive Functions and Motor Skills, and Intensity Differences during and after Immersion," *Virtual Worlds*, vol. 3, no. 1, pp. 64–96, 2024, doi: 10.3390/virtualworlds3010004.

[10] T. Porcino, E. Clua, D. Trevisan, C. N. Vasconcelos, and L. Valente, "Minimizing cybersickness in head-mounted display systems: causes and strategies review,"

in *Proceedings of the 2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, 2020, pp. 154–163, doi: 10.1109/SVR51698.2020.00034.

[11] S. S. Adhanom, N. Lee, H. Folmer, and E. MacNeilage, "Field-of-view restriction to reduce VR sickness in normal and teleoperation modes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 11, pp. 4302–4312, 2021, doi: 10.1109/TVCG.2021.3106497.

[12] R. Islam, S. Ang, and J. Quarles, "Cybersense: A closed-loop framework to detect cybersickness severity and adaptively apply reduction techniques," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2603–2612, 2021, doi: 10.1109/TVCG.2021.3067784.

GLOSSARY

Cybersickness – A form of motion-induced discomfort that occurs when using head-mounted displays, caused primarily by a sensory conflict between visually perceived self-motion and physical vestibular stillness.

Degrees of Freedom (DOF) – The independent translational and rotational axes along which a mechanical structure or virtual body can move; in this system, the physical seat moves in 2-DOF (pitch and roll).

Embodied Interaction – A mode of control where physical user posture, bodily actions, and tangible resistance directly translate into navigation and virtual agency.

FixedUpdate – A synchronized execution loop in the Unity game engine that computes physics calculations and control routines at uniform, fixed time intervals.

Force Sensing Resistor (FSR) – A piezoresistive analog sensor whose electrical resistance decreases with applied mechanical force, used to convert handle-grip pressure into virtual forward speed and climb rate.

Linear Interpolation (Mathf.Lerp) – A mathematical smoothing function that progressively bridges the difference between a current value and a target setpoint over time, used here to simulate flight momentum and prevent abrupt velocity transitions.

Motion Cueing Algorithm (MCA) – A control framework that translates continuous virtual craft movements into physically bounded actuator displacements without exceeding mechanical travel limits.

Pitch – Rotational displacement around a vehicle or seat's lateral horizontal axis, simulating climbing or diving motion.

Roll – Rotational displacement around a vehicle or seat's longitudinal horizontal axis, simulating lateral banking and turning.

Soft-Stop – A firmware-enforced software safety limit that decelerates and stops an actuator before it can strike physical mechanical boundaries.

Tilt Coordination – A motion simulation technique that tilts a user at low angular speeds to align the Earth's gravity vector with perceived virtual acceleration without triggering false semicircular canal cues.

Vection – The illusion of bodily self-motion induced purely through moving visual patterns in the field of view.

Washout Filter – A control filter that returns a motion platform back to its neutral mechanical center position below human sensory thresholds after delivering an initial acceleration cue.

APPENDICES

Appendix A Research Instruments or Extended Results

A.1 Post-Exposure System Evaluation Questionnaire

The post-trial assessment gathered quantitative user perceptions across input responsiveness, motion realism, physical comfort, and cybersickness following flight exposure. Each item was evaluated on a 5-point Likert scale (1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree):

1. **Handle-Grip Throttle Responsiveness**: "Squeezing the handle-mounted sensor to control flight speed and altitude felt natural, predictable, and responded without noticeable lag."
2. **Head-Gyroscope Steering Control**: "Tilting my head to steer and bank the craft felt intuitive and allowed stable navigation without excessive drift."
3. **Motion Cue Realism and Embodiment**: "The physical tilting of the seat matched the acceleration and banking seen in the headset, making the flight feel realistic."
4. **Physical Ergonomics and Fatigue**: "The handle grip resistance remained comfortable throughout the session without inducing noticeable hand or forearm fatigue."
5. **Cybersickness and Disorientation**: "I felt comfortable during the flight and did not experience dizziness, nausea, or disorientation."

A.2 Extended User Response Distribution

Table A.1 details the full distribution of user responses collected across all 35 participants during controlled flight trials.

Evaluation Dimension	Rating 1 (SD)	Rating 2 (D)	Rating 3 (N)	Rating 4 (A)	Rating 5 (SA)	Positive Agreement (%)	Mean Score (out of 5.0)
Grip Throttle Responsiveness	0	1	8	8	18	74.3%	4.2
Head-Tracking Steering Stability	0	3	9	14	9	65.7%	3.8
Vestibular Motion Realism	0	2	6	13	14	77.1%	4.1
Grip Ergonomics (No Fatigue)	2	10	11	9	3	34.3%	2.9
Cybersickness Mitigation	0	2	7	12	14	74.3%	4.1

Table A. 1: Detailed survey response distribution across evaluation dimensions.

A.3 Telemetry Data Frame Specification

Control setpoints are dispatched from Unity to the Arduino/ESP32 microcontroller over a 115200-baud USB serial link at 50 Hz using a structured 9-byte binary packet:

- **Bytes 0–1 (Sync Preamble)**: 0xAA, 0x55 (Frame synchronization markers)
- **Byte 2 (Packet Type)**: 0x01 (Standard motion setpoint command)
- **Bytes 3–4 (Target Pitch)**: Signed 16-bit integer representing target pitch in tenths of a degree ()
- **Bytes 5–6 (Target Roll)**: Signed 16-bit integer representing target roll in tenths of a degree ()
- **Byte 7 (Checksum)**: 8-bit XOR checksum covering bytes 2 through 6 for transmission integrity verification
- **Byte 8 (Frame Terminator)**: 0x0A (Newline delimiter for frame completion)

Appendix B Academic Integrity and AI Use Disclosure

B.1 AI Tools and Usage Scope

In compliance with the SLIIT academic integrity policy and module guidelines for IT4010, generative artificial intelligence tools were utilized in a supportive capacity

during the preparation of this thesis. Table B.1 provides a full disclosure of the tools used, the workflows affected, and the human verification performed.

Tool Name	Version / Model	Purpose of Use	Representative Prompt / Workflow	Human Verification Performed	Sections Affected
Claude / ChatGPT	3.5 Sonnet / GPT-4o	C# Unity syntax assistance and serial buffer debugging	*"Optimize this Unity C# serial write loop to avoid memory allocations and manage baud-rate buffer overflow at 50 Hz."*	All scripts were manually compiled, deployed on hardware, unit tested, and tuned on the physical rig.	Chapter 5
Gemini	1.5 Pro / 2.0 Flash	Grammar refinement, academic tone humanization, and formatting compliance	*"Revise this section to adhere to IEEE standard formatting and SLIIT thesis structure while keeping the language simple, concise, and academic."*	Every paragraph was critically read, verified for factual accuracy against experimental logs, and edited to ensure original authorship.	Chapters 1–8

Table B . 1: Generative AI tools and application disclosure

B.2 Statement of Authorship and Verification

All algorithmic formulations, physical hardware fabrication, sensor calibrations, embedded microcontroller code, experimental flight course designs, and user evaluation trials were independently executed by the author. Generative AI tools were not used to fabricate numerical data, simulate experimental findings, or replace critical analysis. All reported survey metrics, latency figures, and kinematic boundaries reflect primary empirical data collected directly on the experimental motion platform.