ABSTRACT

Title of Thesis: VIBROTACTILE HAPTIC CUEING FOR HUMAN SUPERVISION OF A LEADER–FOLLOWER UAV SWARM IN A SCALED SLALOM TASK

Evan Mitchell Gary
Master of Science, 2026

Thesis Directed by: Professor Umberto Saetti
Department of Aerospace Engineering

This thesis investigates whether vibrotactile haptic feedback can improve human supervision of a leader–follower unmanned aerial vehicle (UAV) swarm during a scaled slalom task. In a leader–follower architecture, the operator directly pilots only the leader vehicle, while follower vehicles maintain prescribed relative positions within the formation. Although this reduces direct manual burden, the operator must still monitor follower motion, formation distortion, and obstacle relationships, which can increase workload and reduce situational awareness, especially in degraded visual environments (DVEs).

To support the operator, a vibrotactile cueing strategy was developed to communicate formation deformation and gate-avoidance information. The primary evaluation was conducted through a simulation-based human-subject experiment using a three-vehicle leader–follower swarm in a scaled ADS-33-inspired slalom course. The simulation framework combined custom quadcopter dynamics, follower formation-control logic, a virtual reality piloting environment, and wearable *bHaptics* devices for

directional cue delivery. Testing was performed in both good visual environment (GVE) and degraded visual environment (DVE) conditions, with and without haptic feedback. Objective metrics included formation-keeping error, missed-gate percentage, speed compliance, and completion time, while subjective evaluation used the Bedford Workload Scale and China Lake Situational Awareness Rating Scale.

Simulation results showed that haptic cueing improved formation-keeping performance, with the strongest gains observed in longitudinal deformation and in lateral deformation under degraded visual conditions. Improvements in missed-gate performance were more modest, and obstacle-navigation success remained strongly influenced by visual condition. Subjective ratings indicated that haptic feedback improved situational awareness without substantially increasing workload. Participant feedback also suggested that haptics were most effective for communicating follower motion and overall swarm geometry.

The work was further extended toward a physical swarm demonstration using custom-built quadcopters, PX4-based control, and a centralized ground-station architecture. Although a full physical experiment was not completed, preliminary testing demonstrated the feasibility of the communication, control, and haptic-integration framework. Overall, the results support vibrotactile cueing as a promising tool for improving human supervision of UAV swarms.

VIBROTACTILE HAPTIC CUEING FOR HUMAN SUPERVISION OF A LEADER–FOLLOWER UAV SWARM IN A SCALED SLALOM TASK

by

Evan Mitchell Gary

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2026

Advisory Committee:
Professor Umberto Saetti, Chair/Advisor
Professor Mumu Xu
Professor Kenneth Yu

© Copyright by
Evan Mitchell Gary
2026

Dedication

I dedicate this work to my grandfather John Briscoe

Acknowledgments

I am extremely grateful to all the people who helped me along this journey, and I greatly enjoyed my experience as a graduate student in the BS/MS program.

First and foremost, I would like to thank my advisor, Professor Umberto Saetti, for giving me the opportunity to engage in research as an undergraduate through the AEROS Scholars program and for setting me on this path. Since then, I have worked with him on several different projects and have learned an inexpressible amount. He has been incredibly helpful at every stage, and I could always count on him to respond and lend a helping hand. It has been an amazing experience to work with and learn from such a knowledgeable and kind individual.

I would also like to thank my coworker on this project, Michael Morcos. Throughout all stages of this work, he has been a mentor in areas I initially knew little about, as well as in areas where I considered myself more experienced. His kindness and willingness to help made every stage of the project much more manageable, and I am extremely thankful for his presence and positive attitude.

This research was supported in part by Lockheed Martin Corporation under agreement no. 311710-00001. I gratefully acknowledge this support, which contributed to the simulation study, experimental development, and hardware implementation.

I am especially thankful for my family and all the support they have provided along the way. To my mother and father—without your encouragement and constant support throughout college, as well as throughout all other stages of life, pursuing my research interests and achieving an M.S. in Aerospace Engineering would not have been possible. I am incredibly grateful for everything you have done for me. To my sister—I am thankful for your positive and kind attitude whenever I visited home, and your continued support while I was at school made it possible to keep going.

My roommates throughout the BS/MS program were also a crucial part of my completion of

this thesis. I would like to express my heartfelt gratitude and appreciation to Joshua Tuttelman, Connor Shields, Colin James, Jeffery Fulep, Adam Brodsky, Ryan Rokhsar, and Joe Monaghan for their friendship and support.

It would be impossible to mention every individual I am thankful for, so to anyone I have left out, I am very sorry.

Thank you, everyone!

Table of Contents

Dedication ii

Acknowledgements iii

Table of Contents v

List of Tables viii

List of Figures ix

List of Abbreviations xi

Chapter 1: Introduction and Background 1
- 1.1 Motivation . . . 1
- 1.2 Problem Statement . . . 2
- 1.3 Research Objective . . . 4
- 1.4 Approach and Scope of Work . . . 4
- 1.5 UAV Swarms and Leader–Follower Control . . . 6
  - 1.5.1 Overview of Multi-UAV Systems . . . 6
  - 1.5.2 Swarm Control Architectures . . . 7
  - 1.5.3 Leader–Follower Formation Control . . . 8
- 1.6 Human Supervision and Haptic Feedback . . . 10
  - 1.6.1 Supervisory Control of Multiple Vehicles . . . 10
  - 1.6.2 Operator Workload, Situational Awareness, and Visual Limitations . . . 10
  - 1.6.3 Haptic Feedback Modalities and Prior Use . . . 12
- 1.7 Mission Task Elements, Handling Qualities, and Scaling . . . 14
  - 1.7.1 ADS-33 Handling Qualities Framework . . . 14
  - 1.7.2 Mission Task Elements and Slalom Tasks . . . 15
  - 1.7.3 Performance Metrics and UAV Scaling . . . 16
- 1.8 Open-Source Autonomy Stacks and Physical Swarm Implementation . . . 18
- 1.9 Research Gap and Thesis Organization . . . 20
  - 1.9.1 Summary of Research Gap . . . 20
  - 1.9.2 Thesis Organization . . . 20

Chapter 2: Simulation Methods 22
- 2.1 Simulation Environment . . . 22
  - 2.1.1 Simulation Software Architecture . . . 22
- 2.2 Flight Simulation Model . . . 24
- 2.3 Vehicle and Swarm Design . . . 25
- 2.4 Flight Control Architecture . . . 26
- 2.5 Pilot Input . . . 29
- 2.6 Visual Display and VR Implementation . . . 29

2.7 Haptic Feedback . . . 31
2.7.1 Formation Deformation Feedback . . . 35
2.7.2 Gate-Avoidance Feedback . . . 36
2.7.3 Cue Rendering and Tactor Activation . . . 38
2.7.4 Cueing Examples . . . 39
2.8 Vehicle Modeling and Scaling Methodology . . . 41
2.9 Mission Task Element Design . . . 44
2.10 Course Geometry and Parameterization . . . 45
2.11 Experimental Breakdown and Design . . . 47
2.12 Performance Metrics . . . 50
2.13 Data Collection and Analysis . . . 51

Chapter 3: Physical Demonstration and Experimental Setup 53
3.1 Quadcopter Components . . . 53
3.1.1 Quadcopter Core Components . . . 53
3.1.2 Airframe Design and Mounting Hardware . . . 54
3.1.3 Propulsion and Power Supply Components . . . 57
3.1.4 Sensors and Receivers . . . 59
3.2 Quadcopter Construction . . . 61
3.3 Flight Stack and Controller . . . 63
3.4 Pilot Input . . . 63
3.5 Flight Modes . . . 64
3.6 High-Level Communication and Control Architecture . . . 65
3.7 Port and Server Flow . . . 67
3.8 Haptic Feedback Interface . . . 69
3.9 Software Architecture . . . 70
3.10 Pre-experiment Simulation and Testing . . . 72
3.11 Experimental Design . . . 73
3.12 Data Collection and Analysis . . . 74
3.13 Limitations of Methods . . . 74

Chapter 4: Results 76
4.1 Simulation Results . . . 76
4.1.1 Example of Slalom Performance . . . 76
4.1.2 Formation-Keeping Performance . . . 77
4.1.3 Gate Avoidance Performance . . . 78
4.1.4 Mental Workload . . . 79
4.1.5 Situational Awareness . . . 80
4.1.6 Participant Feedback . . . 81
4.2 Physical Demonstration Results . . . 83

Chapter 5: Conclusion 88
5.1 Overview of the Research . . . 88
5.2 Summary of Main Findings . . . 88
5.3 Interpretation of Results . . . 89

5.4 Contribution of the Physical Demonstration 89
5.5 Limitations 90
5.6 Recommendations for Future Work 91
5.7 Closing Remarks 92

Appendix A: Subjective Rating Scales 94
A.1 Bedford Workload Scale 94
A.2 China Lake Situational Awareness Rating Scale 94

Bibliography 97

List of Tables

2.1 Froude-scaled target operating conditions for the simulated quadcopter. 43
2.2 Swarm-level geometric quantities used in the scaled slalom course and haptic logic. . . 47

3.1 Port assignments used for leader and follower vehicles in the physical swarm architecture. 69

List of Figures

1.1 Pilot–vehicle system with multimodal perception pathways (adapted from Ref. 1). . . . 12

2.1 High-level simulation software architecture. . . . 23
2.2 Vee formation for quadcopter swarm. . . . 26
2.3 Schematic of the leader closed-loop control architecture. . . . 27
2.4 Follower outer-position tracking loop. . . . 28
2.5 RadioMaster TX16S Mark II radio controller used to provide pilot inputs during the simulation experiment (from Ref. 2). . . . 29
2.6 Varjo Aero VR headset used to provide the immersive first-person visual display during the simulation experiment (from Ref. 3). . . . 30
2.7 Prepar3D heads-up display used during the simulation experiment with the speed circled in red. . . . 31
2.8 Representative pilot views for the two visual conditions used in the simulation experiment. 31
2.9 *bHaptics* devices used to deliver vibrotactile feedback during the simulation experiment: (a) TactSuit X40 haptic vest and (b) Tactosy for Arms armband. . . . 32
2.10 *bHaptics* stimulation mapping used in the simulation experiment, adapted from prior full-body haptic cueing strategies for augmented pilot perception (Refs. 4 and 5). . . . 33
2.11 Representative cueing logic over a single gate-to-gate segment. Longitudinal deformation feedback is active throughout the maneuver, whereas lateral cueing switches from formation-deformation feedback to gate-avoidance feedback during the final approach to a gate. Gate-avoidance cues are generated only when the swarm footprint violates the curved gate boundary. . . . 34
2.12 Illustration of the lateral formation-deformation cueing geometry. Lateral deformation cues are generated only when the follower departs from the lateral no-cue band about the nominal formation offset. . . . 36
2.13 Curved gate boundary used during the gate-avoidance phase. The allowable lateral corridor contracts as the swarm approaches the gate. . . . 37
2.14 Example of lateral deformation cueing. . . . 39
2.15 Example of longitudinal deformation cueing. . . . 40
2.16 Example of gate-avoidance cueing. . . . 40
2.17 Characteristic lengths used for Froude scaling. . . . 41
2.18 Quadcopter model geometry from the vehicle dynamics model. . . . 44
2.19 Original and scaled slalom MTE (Ref. 6). . . . 45
2.20 Overall simulation test setting and procedure. The figure shows the VR-based piloting setup used with Prepar3D, representative GVE and DVE pilot views, and the sequence of experiment phases from setup through debriefing. . . . 48

3.1 Quad Zero carrier board (from Ref. 7). . . . 53
3.2 Control Zero H7 OEM flight controller (from Ref. 7). . . . 54
3.3 Baseline 3DR Quad Zero frame used during initial testing. . . . 55
3.4 Custom PETG-CF frame with circular motor mounts and simplified battery mounting configuration. . . . 56

3.5 GPS mount and riser used to elevate the GPS module above the airframe. 57
3.6 mRo M2-2202 brushless motor (from Ref. 7). 57
3.7 7×5.5 in carbon fiber propeller (from Ref. 7). 58
3.8 mRo 2S 4200 mAh lithium-ion battery pack (from Ref. 7). 59
3.9 SAM GPS module with integrated IST8308 magnetometer (from Ref. 7). 60
3.10 3DR DualBand Wi-Fi radio based on the u-blox ODIN-W260 module (from Ref. 7). . 60
3.11 FrSky R-XSR radio receiver (from Ref. 8). 61
3.12 Fully assembled quadcopter used for the physical swarm demonstration. 62
3.13 Jumper T-Lite radio transmitter used to provide manual inputs to the leader vehicle (from Ref. 9). 64
3.14 High-level communication and control architecture 65
3.15 Per-vehicle MAVLink routing and software-interface flow used in the physical swarm architecture. Each vehicle transmitted telemetry to a dedicated UDP input port on the ground-station computer. MAVProxy then forwarded that stream to a vehicle-specific UDP output for `mavsdk_server` and to a separate UDP output for QGroundControl. The Python swarm code interfaced with each vehicle through the corresponding gRPC endpoint exposed by `mavsdk_server`. 68

4.1 Representative piloted simulation trajectory in the GVE+H condition. 77
4.2 Formation-keeping performance across all experimental conditions, shown using RMS deformation error. Lower values correspond to tighter maintenance of the nominal formation. 79
4.3 Percentage of missed gates across the four experimental conditions. Lower values indicate improved obstacle-avoidance performance. 80
4.4 Subjective evaluation of workload and situational awareness across the four experimental conditions. 81
4.5 Two-vehicle physical test setup used for preliminary leader–follower swarm testing. The leader was manually piloted, while the follower was commanded through PX4 OFFBOARD mode. 84
4.6 Representative trajectory from preliminary physical leader–follower testing. The follower generally tracked the commanded leader-relative motion, with delay attributed primarily to communication and control-loop latency. 85
4.7 Representative video frame from preliminary physical testing at the Fearless Flight Facility. 87

A.1 Bedford Workload Scale used during the simulation experiment. 95
A.2 China Lake Situational Awareness Rating Scale used during the simulation experiment. 96

List of Abbreviations

ADS-33	Aeronautical Design Standard 33
CSV	Comma-separated values
DHA	Direct Haptic Aid
DVE	Degraded Visual Environment
EKF2	Extended Kalman Filter 2
ESC	Electronic speed controller
FRP	Fiber-reinforced plastic
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
gRPC	gRPC Remote Procedure Call
GVE	Good Visual Environment
HUD	Heads-up display
IHA	Indirect Haptic Aid
IMU	Inertial measurement unit
MAVLink	Micro Air Vehicle Link
MTE	Mission Task Element
NDI	Nonlinear dynamic inversion
PD	Proportional-derivative
PID	Proportional-integral-derivative
PX4	Pixhawk 4 autopilot software stack
PWM	Pulse-width modulation
RC	Radio control
RMS	Root-mean-square
SBUS	Serial bus
SDK	Software Development Kit
SITL	Software-in-the-loop

ToF Time-of-flight
TRC Translational-rate-command

UAS Unmanned aerial system
UAV Unmanned aerial vehicle
UDP User Datagram Protocol

VR Virtual reality

Chapter 1: Introduction and Background

1.1 Motivation

UAV swarms consist of multiple aerial vehicles operating cooperatively toward a common objective under a predefined control framework. These systems can range from only a few vehicles to much larger groups, and as the number of vehicles increases, so does the complexity of controlling and supervising the swarm. Despite this added complexity, swarm systems offer several important advantages over single-vehicle operations.

One major advantage of UAV swarms is increased spatial coverage. A group of vehicles distributed across a wider area can provide greater sensing, surveillance, or reconnaissance capability than a single platform alone. Swarm systems also offer redundancy and fault tolerance. If one vehicle malfunctions or is lost during a mission, the remaining vehicles may still be able to continue operating and accomplish the task. In addition, swarm systems can provide greater flexibility than a single larger vehicle. For example, in a mission requiring navigation through obstacles or constrained spaces, a swarm may be able to reconfigure its formation to better suit the environment. Rather than relying on one large vehicle with limited maneuvering options, multiple smaller vehicles can adapt their arrangement to pass through tighter or more complex spaces.

While these characteristics make swarms attractive for many applications, they also introduce significant challenges for the human operator. Instead of monitoring and controlling a single vehicle, the operator must consider the states of multiple agents at once. This includes not only vehicle position and motion, but also formation geometry, overall mission progress, and the health or status of each platform. As the number of vehicles increases, these supervisory demands can quickly become difficult to manage.

The level of operator burden depends in part on the control architecture used. In a leader–follower framework, for instance, the operator directly pilots only one vehicle while the remaining vehicles attempt to maintain prescribed relative positions within the formation. This reduces the need for direct manual control of every drone and makes swarm operation more practical for a single human operator. However, even in this simplified control structure, successful task execution still depends on the behavior of the full swarm. The operator must remain aware of follower motion, formation distortion, and the overall state of the system rather than focusing exclusively on the leader vehicle.

When these supervisory demands are combined with environmental changes, obstacle avoidance requirements, and time-sensitive maneuvering tasks, the result is a highly demanding human–system problem. The operator must simultaneously control the leader vehicle while maintaining awareness of the rest of the swarm, which can increase mental workload and reduce situational awareness. This challenge motivates the need for improved human–swarm interaction methods that can support the operator without adding further demands to the visual control interface.

1.2 Problem Statement

This thesis addresses the demanding human–system problem introduced by supervised UAV swarm operation. Even in relatively simple swarm architectures, the operator must simultaneously account for leader motion, follower positions, overall formation shape, and the relationship between the swarm and nearby obstacles or gates. Presenting all of this information in a form that can be quickly interpreted by the pilot is a significant challenge, particularly when multiple types of information must be understood at the same time.

These demands become more substantial when the operator relies primarily on visual information.

In many swarm-control settings, the pilot must interpret vehicle motion, relative spacing, and path alignment through visual displays alone. As the amount of information increases, it becomes more difficult to process each component individually while also maintaining a coherent understanding of the overall swarm state. This can increase mental workload and make it harder to maintain situational awareness during time-sensitive maneuvers.

The problem becomes even more pronounced in degraded visual environments (DVEs), where visibility is reduced and the quality of visual information is diminished. Conditions such as low light, haze, fog, or other visual obscurants can make it more difficult for the operator to judge vehicle position, formation alignment, and obstacle relationships. In these cases, heavy reliance on the visual channel may limit the operator's ability to effectively supervise the swarm.

Although DVE conditions further highlight the limitations of visual-only feedback, they also motivate the use of supplementary feedback channels. Additional modalities such as auditory or haptic feedback may help redistribute task-relevant information away from the visual channel, reducing visual overload and improving operator understanding of the system state. In particular, vibrotactile feedback offers a promising approach because it can communicate directional or state-based information without interfering with the pilot's visual attention.

Prior work has explored supplementary feedback methods for single-vehicle aerial control, including both auditory and haptic cueing strategies. However, comparatively little research has examined how these ideas extend to human supervision of multi-UAV swarms, where the operator must manage not only vehicle guidance but also formation behavior and collective task execution. This gap motivates the present work. Therefore, there is a need to investigate whether vibrotactile cueing can improve operator performance during human-supervised swarm maneuvers.

1.3 Research Objective

The objective of this thesis is to investigate whether vibrotactile haptic feedback can improve human supervision of a leader–follower UAV swarm during a scaled slalom task. Additional objectives include evaluating the effects of haptic cueing on formation-keeping performance and gate traversal/obstacle avoidance. Pilot workload and situational awareness will also be assessed throughout testing using the Bedford and China Lake scales in order to provide a subjective evaluation of task difficulty. A final objective is to extend this work to a physical demonstration of the swarm slalom task using custom-built quadcopters.

1.4 Approach and Scope of Work

The overall approach of this thesis was to investigate haptic-assisted human supervision of a small leader–follower UAV swarm through both simulation-based testing and a physical demonstration. A leader–follower control architecture was selected because it provides a relatively simple and effective framework for swarm coordination while still preserving the core human–system challenges associated with multi-vehicle supervision. This allowed the study to focus primarily on the effects of haptic feedback rather than on the development of a more complex swarm autonomy strategy.

A vibrotactile feedback strategy was then developed to convey two primary types of task-relevant information to the operator: gate avoidance and formation distortion. Gate-related cueing was intended to support navigation of the swarm through the slalom course, while formation-related cueing was intended to improve operator awareness of follower motion and overall swarm cohesion. Together, these feedback channels were designed to supplement the visual interface and provide the operator

with additional information during task execution.

To evaluate this concept, a scaled slalom task inspired by ADS-33 Mission Task Element principles was implemented as the primary test scenario. The slalom task was selected because it requires simultaneous path-following, obstacle avoidance, speed regulation, and formation maintenance, making it well suited for evaluating human supervision of a swarm. Course geometry and vehicle operating conditions were scaled for the UAV system using Froude-based methods so that the task would remain appropriate for the simulated quadcopter platform.

The primary evaluation of the proposed approach was conducted in simulation. The simulation framework combined the swarm control logic, haptic cueing strategy, and scaled slalom task into a unified experimental environment. Human-subject testing was then performed in order to examine the effects of haptic feedback on formation-keeping performance, gate traversal, workload, and situational awareness. These simulation experiments form the main basis for the results presented in this thesis.

In addition to the simulation study, the work was extended toward a physical demonstration using custom-built quadcopters. This portion of the project focused on implementation feasibility, including vehicle construction, communication architecture, and real-time swarm coordination. While the simulation experiment served as the primary source of formal evaluation, the physical demonstration was used to show that the overall swarm-control concept could be translated to an experimental hardware platform.

1.5 UAV Swarms and Leader–Follower Control

1.5.1 Overview of Multi-UAV Systems

UAV swarms are generally defined as coordinated groups of aerial vehicles working toward a common objective. These systems are motivated by several key advantages over single-vehicle operations, including increased spatial coverage, redundancy and fault tolerance, parallel task execution, and improved operational flexibility. As a result, multi-UAV systems have been applied to a wide range of applications, including surveillance and reconnaissance, search and rescue, environmental monitoring, and military operations such as manned–unmanned teaming (MUM-T) [10, 11, 12].

The primary value of multi-UAV systems lies in their ability to operate cooperatively rather than as independent agents. Through coordination, swarms can distribute tasks, share information, and adapt to dynamic environments more effectively than a single vehicle or a set of independently controlled platforms [13]. However, achieving this level of coordination introduces significant challenges in communication, control, and system design.

Controlling multiple UAVs simultaneously is inherently more complex than operating a single vehicle, particularly as the number of agents increases. This has led to the development of a variety of swarm control architectures and coordination strategies, which aim to balance system autonomy with human oversight. As a result, the design of effective communication and control frameworks remains a central focus in the development of multi-UAV systems.

1.5.2 Swarm Control Architectures

High-level swarm control architectures can generally be categorized into three main classes: centralized, decentralized, and hybrid control [13, 14].

The first is centralized control, which utilizes a central ground station or controller to collect data and relay commands to each individual drone. In this framework, the drones typically do not make independent decisions regarding maneuver selection, but may still collect observations about their environment and transmit this information back to the central node. Because the ground station maintains global knowledge of all vehicle states, centralized control allows for precise trajectory tracking and coordinated motion planning. However, this approach is limited by scalability constraints and introduces a single point of failure in the system [13].

Decentralized control, on the other hand, allows agents to make their own decisions based on their local states, the states of neighboring agents, and the surrounding environment. This approach is particularly useful when the number of agents becomes large or when communication delays and bandwidth limitations inhibit centralized coordination. In fully decentralized systems, the absence of a central controller means that agents do not have direct access to global system information, making tight formation control and precise trajectory tracking more challenging. Instead, control strategies are often behavior-based, relying on local interaction rules such as maintaining neighbor distance, aligning velocities, or moving toward a shared objective [14, 15].

The final class of swarm control is hybrid control, which combines elements of both centralized and decentralized approaches. In this framework, certain responsibilities are delegated to individual agents, while a central ground station or supervisory controller retains authority over higher-level mission objectives. This can take the form of leader–follower architectures, where a designated

leader follows a commanded trajectory and follower agents generate their own control inputs based on relative state information. Most practical UAV swarm implementations adopt hybrid approaches, as they provide a balance between centralized coordination and decentralized robustness [13].

Communication between components of the control architecture is critically important regardless of the control scheme. Each drone must maintain awareness of its state relative to other agents, and ideally its position within a global reference frame. This requires reliable, high-rate communication of state information and commands to ensure timely coordination and stable swarm behavior [13].

1.5.3 Leader–Follower Formation Control

One of the most prevalent formation-control strategies is leader–follower formation control. This control scheme has several variations, each leaning more toward centralized or decentralized implementations; however, the general concept involves direct control of a single vehicle, referred to as the "leader," while the remaining vehicles, or "followers," maintain prescribed positions relative to the leader. Follower commands can either be computed locally or relayed through a ground station, depending on system architecture and mission requirements. The leader–follower approach allows the operator to focus on controlling a single vehicle while the remaining agents operate semi-autonomously. The formation itself is typically defined in the leader reference frame to simplify coordination among follower vehicles [14, 16].

The leader–follower control scheme is attractive for swarm applications for several reasons, starting with its simplicity of implementation. The underlying control laws are relatively straightforward, often relying on position or velocity commands for followers to maintain their nominal states in the leader frame. This approach is also computationally efficient, as follower control inputs can be

determined using only the current state and desired relative position, typically through PID or similar feedback controllers. Additionally, the leader–follower architecture is intuitive for human-supervised systems, allowing operators to quickly understand and adapt to the control structure [14, 17].

The primary control objective of the leader–follower approach is to maintain relative spacing between the leader and follower vehicles. Follower–follower interactions are typically minimized, as the formation is defined entirely within the leader reference frame. Closely related to this objective is the preservation of formation shape, which is generally predefined based on mission requirements and the number of vehicles. Common formation geometries include V-shaped, diamond, and echelon configurations, each offering different trade-offs in visibility, efficiency, and collision avoidance [18]. Additional objectives include accurate tracking of the leader trajectory and avoidance of inter-vehicle collisions. Leader trajectory tracking is typically achieved using onboard state estimation (e.g., IMU and GPS), with this information relayed to follower agents. Collision avoidance is often addressed through appropriate formation spacing, safety margins, and, in some cases, additional sensing or local avoidance strategies [15].

Although the leader–follower technique is highly effective, it is not without limitations. A major challenge is its sensitivity to leader state estimation errors. Because follower commands are derived from the estimated leader state, any inaccuracies in that estimate can propagate through the formation, resulting in distorted geometries or, in severe cases, inter-vehicle collisions. This phenomenon is commonly referred to as error propagation in formation control systems [16]. These effects can be mitigated through improved sensing accuracy, higher communication update rates, and the integration of additional sensing modalities such as relative positioning or collision detection systems. Overall, the effectiveness of the leader–follower architecture depends heavily on communication reliability, sensor quality, and system update frequency.

1.6 Human Supervision and Haptic Feedback

1.6.1 Supervisory Control of Multiple Vehicles

Control of multi-UAV systems is typically implemented using a supervisory control framework, in which the human operator oversees the behavior of multiple autonomous vehicles rather than directly controlling each vehicle at all times. In this paradigm, low-level flight stabilization and navigation are handled by onboard control systems, while the human operator provides higher-level commands such as waypoint selection, task assignment, or mission objectives. This structure allows a single operator to manage multiple aerial vehicles simultaneously by shifting the human role from continuous manual control to monitoring, decision-making, and intervention when necessary.

In swarm operations, supervisory control requires the operator to maintain awareness of the state of multiple agents while ensuring that the overall mission objectives are satisfied. Rather than issuing continuous control inputs to individual vehicles, operators typically interact with the system through higher-level commands that influence group behavior, formation structure, or trajectory planning. Studies of UAV swarm interfaces have shown that effective supervisory control depends heavily on interface design and the operator's ability to interpret system information quickly and accurately [19]. Decision-support tools and interface architectures have therefore been proposed to assist operators in monitoring system state and making timely interventions during complex multi-UAV missions [20].

1.6.2 Operator Workload, Situational Awareness, and Visual Limitations

Supervising multiple UAVs simultaneously introduces significant cognitive demands on the human operator. In addition to monitoring the status of individual vehicles, operators must track

mission progress, evaluate environmental conditions, and maintain awareness of the spatial relationships between vehicles within the swarm. These responsibilities require continuous attention and information processing, which can rapidly increase operator workload as the number of vehicles or mission complexity grows [21].

A key factor influencing operator performance in these systems is situational awareness, which refers to the operator's ability to perceive relevant system states, comprehend their meaning, and anticipate future system behavior. Maintaining situational awareness becomes increasingly difficult in multi-UAV systems because operators must interpret information from multiple vehicles simultaneously while also monitoring mission-level objectives [10]. Studies of swarm interface design have shown that increases in vehicle count or information density can quickly degrade operator awareness and decision-making performance if the interface does not effectively organize or present system information [19]. As a result, managing operator workload and preserving situational awareness remain central challenges in the design of human–swarm interaction systems.

It is common in the literature to refer to normal visual conditions as a good visual environment (GVE) and poor visual conditions (e.g., nighttime or low-visibility weather) as a degraded visual environment (DVE). Most modern UAV control interfaces rely primarily on visual displays, meaning that in DVE conditions operators may struggle to maintain effective control and situational awareness of the vehicle. This issue becomes more significant when supervising multiple vehicles in a swarm formation, as the operator must simultaneously monitor several vehicle states and trajectories, further increasing cognitive workload and information density [10]. Figure 1.1 conceptually illustrates how the pilot's control actions are shaped by perceived error and multimodal sensory information, including visual, auditory, haptic, and vestibular channels.

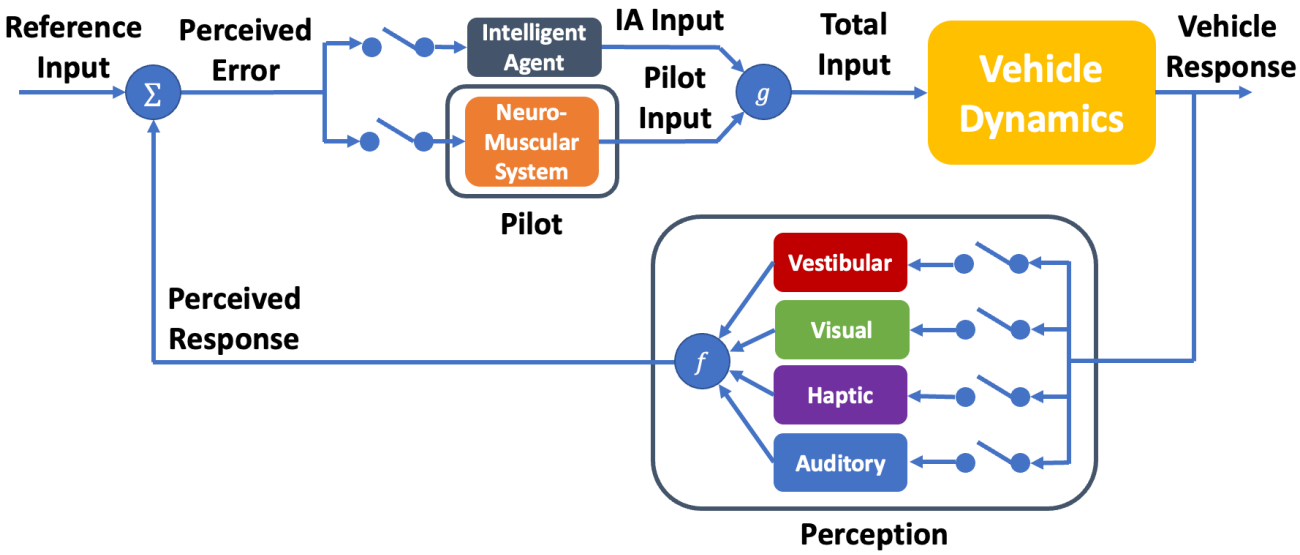


Figure 1.1: Pilot–vehicle system with multimodal perception pathways (adapted from Ref. 1).

Several studies have investigated interface designs and interaction methods aimed at reducing operator cognitive workload under these conditions [21]. Other work has explored the use of additional feedback systems to supplement visual information, including auditory and haptic cueing methods [1, 22]. These approaches aim to distribute information across multiple sensory channels, allowing operators to maintain better situational awareness without further increasing the demands placed on visual attention.

1.6.3 Haptic Feedback Modalities and Prior Use

Haptic feedback used in human–machine interfaces is generally broken into two distinct types: kinesthetic (force feedback) and tactile (cutaneous feedback). Kinesthetic feedback utilizes the user's muscles and joints to send forces and motion. This is typically done through robotic devices such as force-feedback joysticks or exoskeletons. Tactile feedback, on the other hand, stimulates skin receptors and can be broken down further into pressure, vibration, and skin deformation subtypes

[23]. Vibrotactile feedback is one of the most widely used tactile feedback modalities in wearable systems and human–machine interfaces because of its simplicity and ease of implementation [24].

Vibrotactile feedback is very relevant for this study due to its need for lightweight and simple integration with the experimental system. Vibrotactile feedback also does not interfere with control inputs, making the technique a suitable option for vehicle control. Using vibrotactile feedback also allows the user to encode directional information [25]. By assigning a reference frame, specific tactors or groups of tactors can correspond to specific directions within that frame. From there, either direct haptic aid (DHA) or indirect haptic aid (IHA) strategies can be used to determine appropriate control actions based on the directional cues. DHA provides cues that indicate the correct control direction, allowing the pilot to make corrective maneuvers in the direction of the stimulus. IHA, on the other hand, is meant to inform pilots of their surroundings so that they move in the opposite direction of the cue [26]. For this experiment, a combination of IHA and DHA was implemented to achieve both guidance and situational awareness.

Haptic feedback has proven exceptionally useful for improving teleoperation and remote manipulation systems in several fields, such as robotic surgery, telemanipulation, and remote robotics. Haptic feedback has been widely studied in teleoperation systems to improve operator awareness and control precision [27]. Studies have shown that haptic feedback can improve operator situational awareness in applications where environmental perception plays a critical role. Task error can also be reduced through direct haptic aid (DHA) strategies, as reference cues guide operator actions along a desired path in high-risk scenarios such as robotic surgery [28]. In other applications, haptic feedback in the form of kinesthetic force feedback allows interaction forces to be reflected to the operator, enabling more precise manipulation of remote systems [29]. These findings demonstrate that haptic feedback can significantly improve operator awareness and control precision in remote manipulation tasks,

motivating its investigation for other human–machine interaction domains such as aerial vehicle control.

The relevance of haptic feedback techniques extends beyond robotics into aerial vehicle control. Previous studies have shown that haptic cues can assist pilots in tracking tasks and improve performance, particularly in degraded visual environments (DVE), where visual information may be limited or unreliable [1]. In many cases, these haptic signals are combined with other feedback modalities such as spatial audio or visual cues to further aid pilot performance during these tracking tasks [4]. More recent studies have explored the integration of wearable haptic systems in full-vehicle simulation experiments, including hover-based and slalom Mission Task Elements (MTEs) representative of operational flight tasks [5]. These studies demonstrated that haptic cueing can improve pilot tracking performance in DVE conditions while also reducing cognitive workload and increasing pilot situational awareness.

Although significant research has examined the challenges of supervising multiple UAVs, relatively little work has explored how haptic feedback could assist operators in managing swarm behavior. Prior studies of UAV interface design have primarily focused on improving visual displays or decision-support tools for swarm supervision [19, 21]. However, these approaches still rely heavily on visual information, which can become overloaded as swarm size and mission complexity increase. This gap in the existing literature motivates the research presented in this thesis.

1.7 Mission Task Elements, Handling Qualities, and Scaling

1.7.1 ADS-33 Handling Qualities Framework

The Aeronautical Design Standard ADS-33 establishes handling qualities requirements for military rotorcraft, providing a structured framework for evaluating vehicle controllability and pilot workload

across a range of operational scenarios [6]. The standard defines quantitative performance criteria and evaluation procedures intended to ensure that rotorcraft can be safely and effectively operated in demanding mission environments. Rather than relying solely on analytical metrics, ADS-33 emphasizes task-based evaluation through representative flight maneuvers that reflect realistic operational demands. These maneuvers allow pilot–vehicle system performance to be assessed in terms of both vehicle response characteristics and pilot workload while completing specific mission objectives. As a result, ADS-33 has become a widely used benchmark for assessing rotorcraft handling qualities and guiding the design and evaluation of flight control systems [30, 31].

Within ADS-33, vehicle performance is evaluated through a set of standardized Mission Task Elements (MTEs) that represent common operational maneuvers such as precision hover, lateral repositioning, and obstacle avoidance. Each task is designed to challenge specific aspects of vehicle handling, including stability, control responsiveness, and pilot workload during maneuver execution. Performance is assessed using both objective measures, such as path tracking accuracy and speed regulation, and subjective pilot workload ratings. By grounding vehicle evaluation in representative operational tasks, the ADS-33 framework provides a practical methodology for assessing pilot–vehicle system performance under realistic flight conditions.

1.7.2 Mission Task Elements and Slalom Tasks

Mission Task Elements (MTEs) form the core evaluation methodology within the ADS-33 handling qualities framework. MTEs are standardized flight tasks designed to replicate representative operational maneuvers encountered during rotorcraft missions [6]. Rather than evaluating aircraft performance solely through isolated response metrics, MTEs assess the combined performance of the pilot and

vehicle while executing realistic flight maneuvers. Each task is structured to challenge specific aspects of vehicle controllability, including stability, responsiveness, and pilot workload during maneuver execution.

Typical MTEs include tasks such as precision hover, lateral repositioning, vertical maneuvers, and obstacle navigation. During these tasks, performance is evaluated using both objective measures and pilot workload assessments. Objective metrics often include path tracking accuracy, adherence to speed constraints, obstacle avoidance, and task completion time, while pilot workload ratings provide insight into the perceived difficulty of controlling the vehicle. By evaluating performance within the context of realistic operational maneuvers, MTEs provide a practical framework for assessing the effectiveness of flight control systems and pilot–vehicle interaction.

The slalom task is a Mission Task Element outlined in the ADS-33 handling qualities guidelines [6] and is commonly used for rotorcraft handling qualities evaluation. The task consists of alternating lateral turns along a straight runway, thereby testing lateral control, speed regulation, and path-following performance of the pilot–vehicle system. In many experimental implementations, cones or other markers are used to indicate the outer bounds of each turn. However, the gated slalom variation uses pylons on both sides of the path to form physical gates that must be traversed during the maneuver. The gated slalom was utilized in this study because the defined gate boundaries allow for more precise evaluation of path-following performance and maneuver execution during swarm control experiments.

1.7.3 Performance Metrics and UAV Scaling

Based on ADS-33 guidelines, all turns must be completed within the defined gate boundaries. Any collision with a gate or a missed gate is considered a failure for a given trial [6]. Target speeds

are defined under both good visual environment (GVE) and degraded visual environment (DVE) conditions for both adequate and desired performance levels. Although completion time is not explicitly defined as a primary performance metric in ADS-33, it is commonly recorded during experimental evaluations because it directly reflects the pilot's ability to maintain the desired speed profile while executing the maneuver [32].

While ADS-33 was originally developed for full-scale rotorcraft, recent research has explored the adaptation of MTE-based evaluation methods to unmanned aerial systems (UAS). In particular, proposed UAS-specific MTEs have been developed to account for the different dynamic characteristics and operational roles of small multirotor platforms [32]. Related work on multirotor system identification and dynamic scaling has further demonstrated the feasibility of applying rotorcraft handling qualities evaluation techniques to smaller aerial vehicles [33]. These efforts demonstrate that task-based evaluation methodologies remain valuable for assessing the handling qualities and control performance of smaller aerial vehicles, particularly when evaluating advanced control interfaces or human–vehicle interaction strategies.

Many studies have explored the application of MTEs defined in ADS-33 to smaller unmanned aerial vehicles. However, the course geometries and maneuvering requirements defined for full-scale rotorcraft are often unsuitable for small multirotor platforms due to significant differences in vehicle size, mass, and dynamic response characteristics. To address these differences, dynamic similarity techniques are commonly used to translate full-scale maneuver requirements to smaller vehicles. One widely used approach is Froude scaling, which preserves the ratio of inertial forces to gravitational forces between geometrically similar systems. By maintaining this similarity relationship, maneuver geometries and speed requirements can be scaled in a way that preserves comparable vehicle dynamics across different vehicle sizes [33, 34].

The same scaling factor used for individual vehicle metrics is often applied to the geometry of the flight task as well. Course dimensions such as gate spacing and lateral displacement are scaled to maintain geometric similarity between the reference maneuver and the scaled experiment [6, 32]. Complications arise, however, when multiple vehicles are used in mission task elements such as the slalom (see 2.9). If the course geometry is scaled for a single small-scale vehicle but multiple vehicles are flown simultaneously, the resulting course may become too constrained for coordinated swarm motion. This limitation becomes particularly relevant in swarm control studies, where multiple vehicles must simultaneously complete scaled maneuvers while maintaining formation structure and collision-free separation.

1.8 Open-Source Autonomy Stacks and Physical Swarm Implementation

The development of physical multi-UAV systems has been significantly enabled by the maturation of open-source autopilot frameworks and standardized communication protocols. Among these, the MAVLink protocol has emerged as a widely adopted lightweight messaging system for communication between unmanned vehicles, ground control stations, and external computing platforms. MAVLink is specifically designed for low-latency, bandwidth-constrained environments and supports structured message exchange for state telemetry, command transmission, and system health monitoring [35, 36].

The PX4 autopilot has become one of the most commonly used open-source flight control systems for research applications due to its modular architecture and support for external control interfaces. In particular, PX4's OFFBOARD mode enables real-time trajectory control from an external computing source, allowing vehicle motion to be driven by algorithms running outside the flight controller. This paradigm has been widely used in multi-agent and swarm robotics research, where

centralized or distributed controllers generate commands based on shared state information [37, 38].

Centralized control architectures, in which a ground-based computer computes and distributes commands to multiple vehicles, are frequently employed in experimental swarm systems due to their simplicity and reduced onboard computational requirements. Such architectures have been demonstrated in formation flight, cooperative transport, and trajectory tracking applications, where external controllers process global state information and enforce coordination constraints across the swarm [39, 40]. These approaches are particularly suitable for small UAV platforms that lack the onboard processing capability required for fully decentralized autonomy.

In practical implementations, middleware tools are required to interface with MAVLink-based systems and manage communication between multiple vehicles and control processes. MAVProxy is commonly used as a lightweight MAVLink routing and forwarding tool, enabling telemetry streams from individual vehicles to be distributed to multiple endpoints, including ground control stations and external autonomy software. This capability is critical in multi-UAV experiments, where telemetry must be simultaneously visualized, logged, and processed for control purposes.

At a higher level, software development kits such as MAVSDK provide structured APIs for interacting with MAVLink-enabled vehicles, allowing developers to implement control logic without directly handling low-level message parsing. MAVSDK-based workflows have been adopted in numerous research and prototyping applications due to their ability to simplify command generation and state monitoring while maintaining compatibility with PX4 systems.

Together, these technologies form a cohesive software stack for physical swarm implementation. MAVLink provides the communication backbone, PX4 enables stable flight and external control through OFFBOARD mode, MAVProxy supports flexible telemetry routing, and MAVSDK facilitates high-level interaction with the vehicles. This combination enables the realization of centralized leader–

follower architectures in which a ground-station computer collects state information, computes formation control objectives, and transmits real-time commands to follower vehicles.

1.9 Research Gap and Thesis Organization

1.9.1 Summary of Research Gap

Existing studies have shown that supervisory swarm control places substantial demands on the operator, particularly when task-relevant information must be interpreted visually and in real time. Prior research has also shown that haptic feedback can improve pilot performance and awareness in single-vehicle control tasks, especially in degraded visual conditions. However, comparatively little work has combined these two directions by studying vibrotactile cueing for human-supervised multi-UAV swarm control within a structured, task-based evaluation framework. Furthermore, relatively little work has examined swarm haptic cueing using an ADS-33-inspired scaled slalom task that explicitly evaluates both obstacle traversal and formation-keeping behavior. The present work addresses this gap through a simulation-centered study and an extension toward physical swarm implementation.

1.9.2 Thesis Organization

The remainder of this thesis is organized as follows. Chapter 2 presents the simulation methods, including the virtual environment, swarm architecture, haptic cueing strategy, vehicle modeling, scaling methodology, and human-subject experimental design. Chapter 3 presents the physical demonstration and experimental setup, including the quadcopter hardware, communication architecture, assembly process, and physical implementation details. Chapter 4 presents the results of the simulation study and the physical demonstration. Chapter 5 summarizes the main conclusions of the work, discusses

limitations, and provides recommendations for future research.

Chapter 2: Simulation Methods

2.1 Simulation Environment

All simulation-based experiments were conducted using Prepar3D as the primary virtual environment. Prepar3D was selected due to its high-fidelity physics engine, extensibility through its Software Development Kit (SDK), and compatibility with externally defined vehicle dynamics, control laws, and experimental interfaces. Custom add-ons were developed and integrated through the Prepar3D SDK to support real-time simulation, haptic feedback, and experiment execution.

2.1.1 Simulation Software Architecture

The simulation framework was composed of several interacting software components responsible for experiment management, vehicle dynamics and control, follower position tracking, haptic cue generation, and graphical rendering. Together, these components supported real-time execution of the slalom task and the associated human–swarm interaction experiment. At a high level, pilot inputs were applied to the leader vehicle, the resulting vehicle states were propagated through the flight dynamics model, follower position commands were generated to maintain the desired swarm geometry, and haptic cues were computed from the resulting formation and gate-related errors. The simulated vehicle and environment states were then rendered in Prepar3D for the operator.

A description of the major software components is provided below.

Simulation Wrapper. The simulation wrapper integrated the major modules required to execute the experiment. It facilitated communication among those modules and ensured that the required data were recorded for post-processing analysis.

Augmentation System. The augmentation system contained the leader-vehicle control laws, including

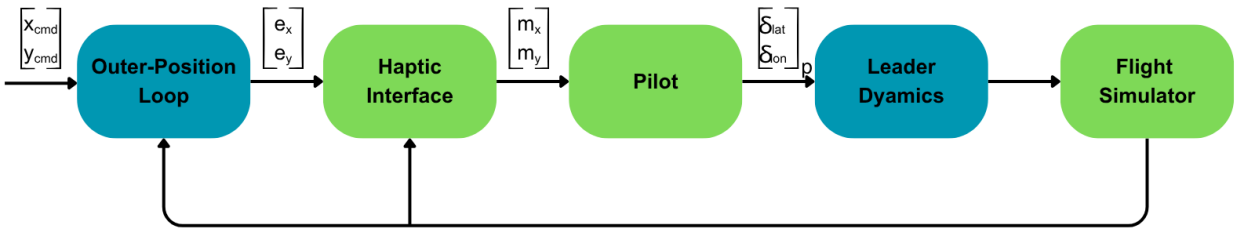


Figure 2.1: High-level simulation software architecture.

the translational-rate-command response and altitude-hold behavior. It also incorporated trim states and trim control inputs, with interpolation performed as a function of forward speed.

Flight Simulation Model. The flight simulation model contained the stitched residualized vehicle dynamic model. It processed the augmented control inputs to compute vehicle states and generate the corresponding outputs used throughout the simulation framework.

Outer-Position Loop. The outer-position loop received external position commands and generated the required attitudes to track those commands. In the present work, this loop was used to support follower formation tracking relative to the piloted leader.

Haptic Interface. The haptic interface calculated the haptic cues for each control axis and transmitted the corresponding signals to the *bHaptics* devices. It also determined the magnitude and direction of each cue and mapped the feedback to the appropriate body location.

Graphical Environment. The graphical environment used Prepar3D® for visual rendering, with the main camera attached to the front of the leader vehicle to provide a first-person piloting view. A Virtual

Reality (VR) plugin was also used to display the scenario in VR goggles.

2.2 Flight Simulation Model

The vehicle dynamics used in this study were generated using an in-house MATLAB/Simulink multirotor simulation framework rather than the default Prepar3D vehicle model. Within this framework, the aircraft fuselage was represented as a rigid body and the corresponding aerodynamic forces and moments were computed using simplified aerodynamic representations suitable for real-time piloted simulation. The model architecture also allowed rotor placement, vehicle geometry, and control-law integration to be defined externally and then incorporated into the simulation environment.

At a general level, the simulated vehicle dynamics may be expressed as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \qquad \mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}), \tag{2.1}$$

where $\mathbf{x}$ is the vehicle state vector, $\mathbf{u}$ is the control input vector, and $\mathbf{y}$ contains the outputs used by the simulation and control architecture. In the present work, the retained states of primary interest included the rigid-body translational velocities, angular rates, Euler attitudes, and inertial positions required for piloted maneuvering and follower tracking.

For use in real-time simulation, the nonlinear vehicle model was trimmed and linearized over the relevant operating envelope and subsequently converted into a stitched residualized form. This provided a computationally efficient representation of the vehicle dynamics while preserving the dominant closed-loop response characteristics required for piloted evaluation. The resulting model was then integrated with the simulation wrapper and control architecture to provide the vehicle states used for visualization, haptic cue generation, and follower tracking logic.

The simulated vehicle was based on a quadcopter configuration visually represented in Prepar3D using an Indago-like asset. Although that visual model was used for graphical presentation, the underlying dynamics, trim states, and control response were entirely determined by the external vehicle model. A representative view of the modeled quadcopter geometry is shown in Fig. 2.18.

An important modeling decision in this simulation was that the follower vehicles were not represented through detailed external visual models. Instead, follower behavior was captured through the formation-control logic and the associated relative-position states used for haptic cue generation and performance assessment. This choice was appropriate because the pilot's viewpoint was attached to the front of the leader vehicle and the slalom task was conducted primarily in the forward field of view. As a result, the experiment emphasized operator awareness of swarm behavior through the control and cueing framework rather than through direct visual observation of the follower vehicles.

2.3 Vehicle and Swarm Design

The leader vehicle was visually modeled after the Indago 3 UAS, which was already available within the Prepar3D asset library. This allowed development effort to be focused on control logic, haptic feedback, and experimental design rather than vehicle visualization. Although the Indago visual model was used, the default Prepar3D vehicle dynamics were fully replaced with custom dynamics generated from the in-house simulation framework. The follower vehicles were not assigned detailed visual models, as the operator remained forward-facing throughout the task and direct follower visualization was not required for task execution.

A leader–follower swarm control architecture was employed for the simulation study. The swarm consisted of three agents arranged in a V-shaped formation, with the leader at the front and

two wingmen trailing symmetrically behind (see Fig. 2.2). The human operator directly controlled only the leader vehicle, while the follower vehicles attempted to maintain prescribed relative positions with respect to the leader. Although pilot inputs were applied to only a single vehicle, task success depended on the coordinated behavior of the entire swarm.

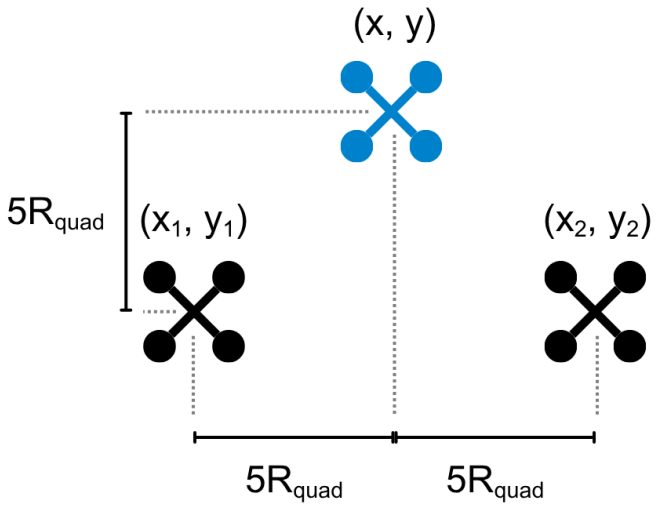


Figure 2.2: Vee formation for quadcopter swarm.

2.4 Flight Control Architecture

The flight control architecture was designed to provide a stable and predictable piloted response for the leader vehicle while simultaneously supporting position tracking for the follower vehicles. For the leader, the augmentation system was based on a nonlinear dynamic inversion (NDI) framework configured around a translational-rate-command (TRC) response type. This structure allowed the operator to command the vehicle primarily through pitch and roll inputs while the control system shaped the corresponding translational response. In the vertical axis, an altitude-hold function was

included to reduce pilot workload and eliminate the need for continuous throttle management during the slalom task.

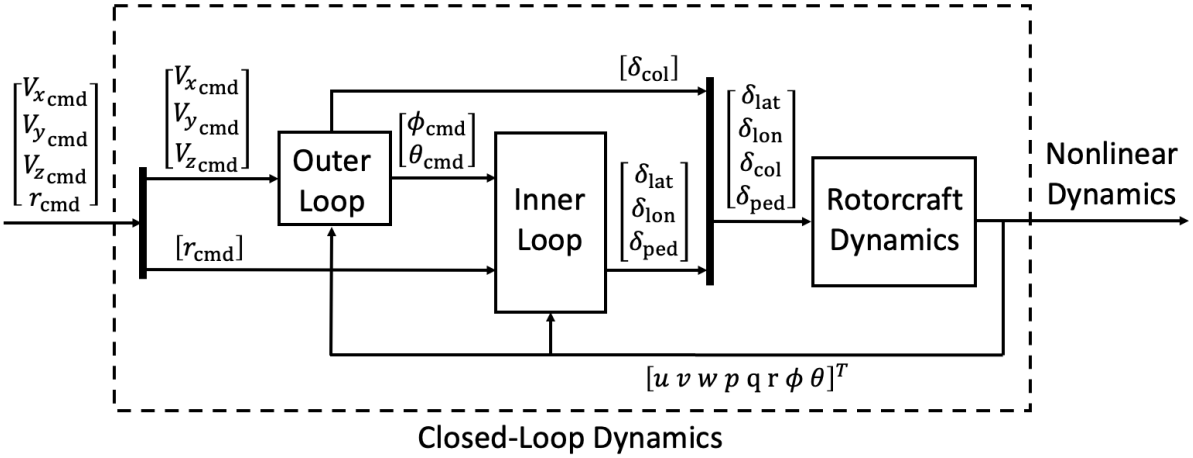


Figure 2.3: Schematic of the leader closed-loop control architecture.

The leader control structure was organized hierarchically, with inner-loop stabilization and response shaping operating together with outer-loop command behavior. In practice, this allowed the pilot to interact with the vehicle through a more manageable and well-conditioned control response than would be obtained from the raw vehicle dynamics alone. As a result, the operator could focus on maneuvering through the slalom course and responding to haptic cues rather than on low-level stabilization. In the present study, roll and pitch were shaped to provide an attitude-command/attitude-hold type response, while yaw was treated with a rate-command/attitude-hold response.

The follower vehicles were controlled separately from the piloted leader. Desired follower positions were defined relative to the leader, and an outer-position tracking loop was used to drive the followers toward those commanded locations. A representative form of the reduced-order follower model may be written as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \qquad \mathbf{y} = \mathbf{C}\mathbf{x}, \tag{2.2}$$

where $\mathbf{x}$ contains the reduced set of translational velocities and inertial positions, and $\mathbf{u}$ contains the

commanded attitudes and collective-like input used to regulate motion. The corresponding dynamic-inversion outer-loop control law was implemented in the form

$$\mathbf{u} = (\mathbf{CAB})^{-1}\left(\boldsymbol{\nu} - \mathbf{CA}^2\mathbf{x}\right), \tag{2.3}$$

where $\boldsymbol{\nu}$ is the pseudo-control input vector.

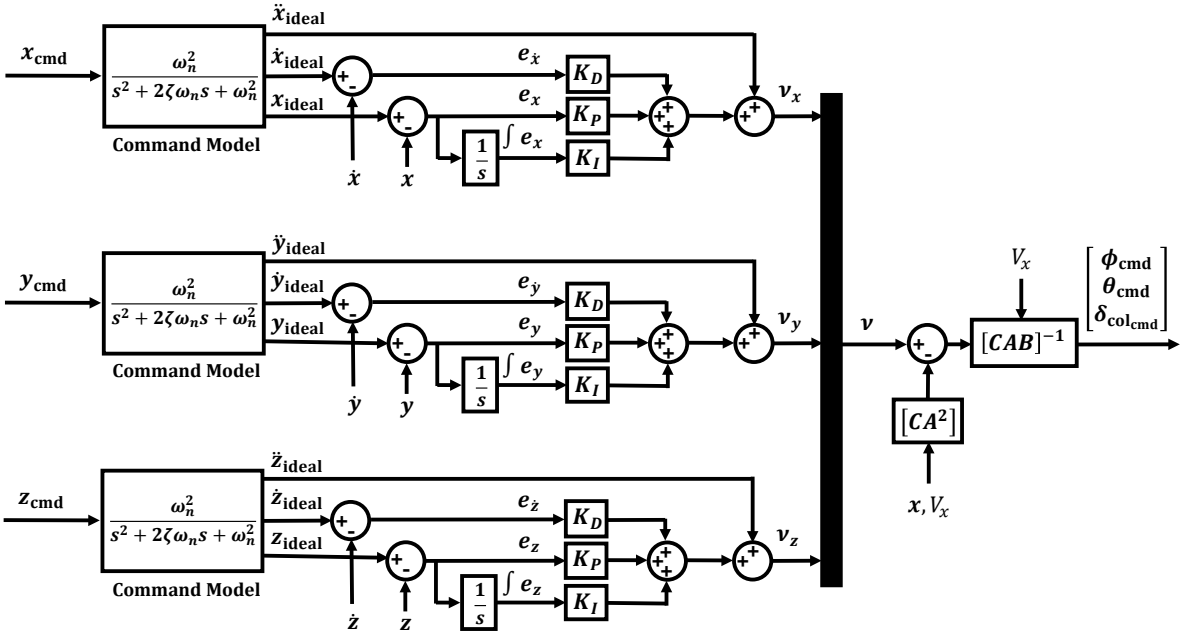


Figure 2.4: Follower outer-position tracking loop.

The pseudo-control input was constructed from the reference trajectory and the corresponding position and velocity tracking errors according to

$$\boldsymbol{\nu} = \ddot{\mathbf{r}}_{\mathrm{cmd}} + \mathbf{K}_D\dot{\mathbf{e}} + \mathbf{K}_P\mathbf{e} + \mathbf{K}_I\int \mathbf{e}\,dt, \tag{2.4}$$

where $\mathbf{e}$ denotes the inertial-frame tracking error vector. This structure enabled the followers to maintain the desired formation geometry while the leader executed the maneuver. In this way, the

overall control architecture combined piloted leader motion, autonomous follower tracking, and haptic cue generation within a unified closed-loop simulation framework.

2.5 Pilot Input

Pilot inputs were provided using a Radiomaster TX16S radio transmitter, which was wired directly to the host desktop computer and interfaced with the Prepar3D environment (Fig. 2.5). To simplify task execution and reduce confounding effects, the throttle channel was disabled and the vehicle operated in an altitude-hold mode. Pilots were also discouraged from providing yaw inputs, leaving pitch and roll as the primary control axes for maneuvering through the slalom course.



Figure 2.5: RadioMaster TX16S Mark II radio controller used to provide pilot inputs during the simulation experiment (from Ref. 2).

2.6 Visual Display and VR Implementation

The operator used Varjo Aero VR goggles during the simulation experiment, allowing the task to be experienced from an immersive first-person viewpoint. The camera was attached to the front of the leader drone using the default Prepar3D camera attachment behavior, thereby presenting the course

from the leader vehicle's perspective. This gave the pilot a forward-looking viewpoint consistent with the motion of the piloted vehicle throughout the maneuver.



Figure 2.6: Varjo Aero VR headset used to provide the immersive first-person visual display during the simulation experiment (from Ref. 3).

Although the camera heading was not artificially fixed, pilots were instructed to maintain a predominantly forward-looking view during the task because the relevant course features and required maneuvering cues were located ahead of the leader vehicle. In this way, the visual scene remained operationally similar to a forward-looking pilot view even though small head and camera rotations were still possible.

A Prepar3D integrated heads-up display (HUD) was also used during the experiment. This display provided attitude and speed information to the pilot, allowing comparison between the instantaneous vehicle state and the target speed requirements associated with the slalom task, as outlined in Section 2.9.

Two visual conditions were implemented in the simulation: a good visual environment (GVE) and a degraded visual environment (DVE). The GVE condition was represented using daytime lighting and clear-sky weather. The DVE condition was represented using nighttime lighting, fog, and additional brightness reduction to further limit visual scene clarity. These visual manipulations were intended to

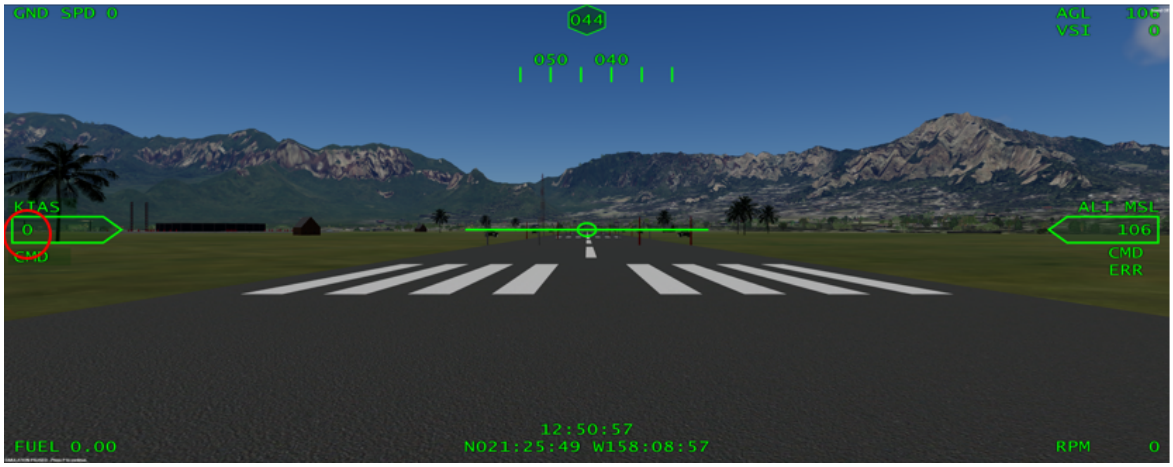
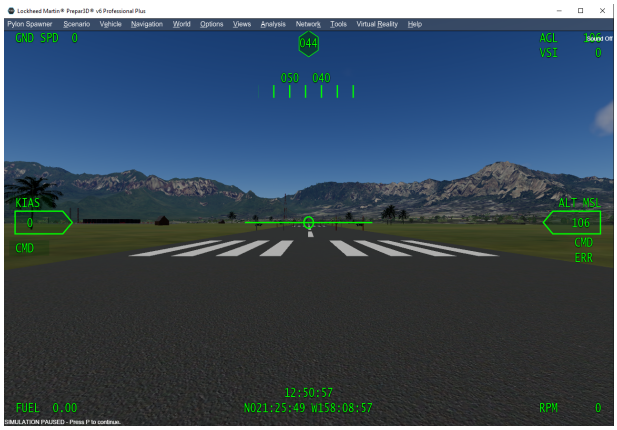


Figure 2.7: Prepar3D heads-up display used during the simulation experiment with the speed circled in red.

change the quality of the forward scene while leaving the underlying flight task and swarm-control logic unchanged.



(a) GVE: daytime and clear-sky condition.



(b) DVE: nighttime, fog, and reduced-brightness condition.

Figure 2.8: Representative pilot views for the two visual conditions used in the simulation experiment.

2.7 Haptic Feedback

Haptic feedback was provided primarily along the pitch and roll axes using commercial *bHaptics* devices. Pitch-related cues were rendered through a *bHaptics* vest, while roll-related cues were

delivered through *bHaptics* armbands, as shown in Fig. 2.9. In the implemented mapping, pitch cues conveyed longitudinal swarm-spacing information and roll cues conveyed lateral swarm information, including both formation deformation and gate-avoidance guidance.

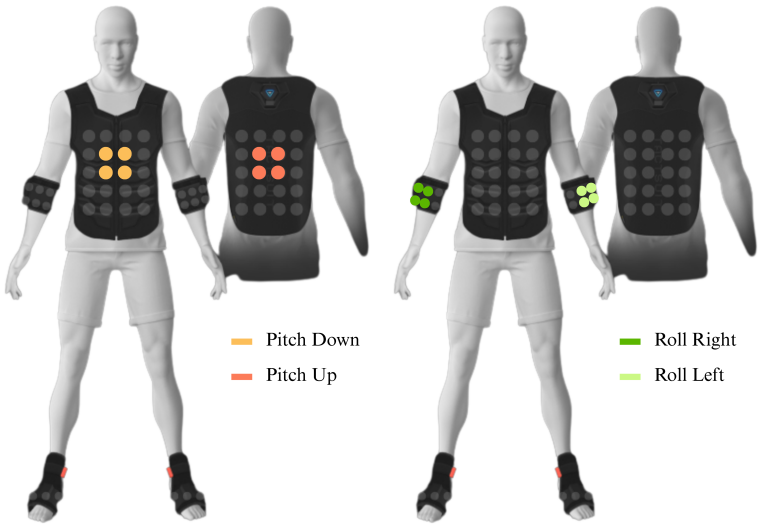


(a) TactSuit X40 haptic vest (from Ref. 41).

(b) Tactosy for Arms armband (from Ref. 42).

Figure 2.9: *bHaptics* devices used to deliver vibrotactile feedback during the simulation experiment: (a) TactSuit X40 haptic vest and (b) Tactosy for Arms armband.

The cueing logic was designed to be directionally consistent with the corrective action required of the pilot. As shown in Fig. 2.10, a vibration on the chest indicated a need for forward pitch input, whereas a vibration on the back indicated a need to reduce forward pitch. Likewise, stimulation of the left arm indicated a left roll correction and stimulation of the right arm indicated a right roll correction. This directional haptic alignment was intended to reduce interpretation time and support intuitive pilot response.



(a) Pitch. (b) Roll.

Figure 2.10: *bHaptics* stimulation mapping used in the simulation experiment, adapted from prior full-body haptic cueing strategies for augmented pilot perception (Refs. 4 and 5).

Two distinct lateral information sources were encoded during the task: formation deformation and gate avoidance. Longitudinal deformation feedback remained active throughout the maneuver, while the lateral cueing mode depended on the vehicle's position within the gate-to-gate segment. A representative cueing timeline over one course segment is illustrated in Fig. 2.11. In that logic, lateral formation-deformation cueing was used away from the gate, whereas gate-avoidance cueing governed the final approach to the next gate.

For each cue channel, the underlying haptic command was computed from a proportional-derivative (PD) relationship of the form

$$M_i = K_{\mathrm{P},i} e_i + K_{\mathrm{D},i} \dot{e}_i, \qquad i \in \{x, y\}, \tag{2.5}$$

where e_i is the relevant error quantity, $\dot{e}_i$ is its time derivative, and M_i is the corresponding continuous-

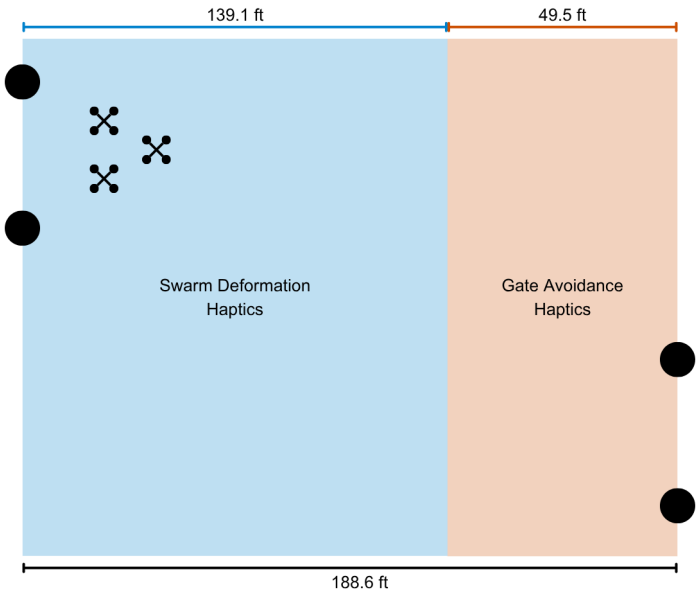


Figure 2.11: Representative cueing logic over a single gate-to-gate segment. Longitudinal deformation feedback is active throughout the maneuver, whereas lateral cueing switches from formation-deformation feedback to gate-avoidance feedback during the final approach to a gate. Gate-avoidance cues are generated only when the swarm footprint violates the curved gate boundary.

valued cue command. In the implemented *bHaptics* interface, this command was then scaled and discretized for actuator rendering according to

$$a_i = \lceil 100\, M_i \rceil\,, \tag{2.6}$$

where a_i denotes the integer intensity level sent to the corresponding tactors. Thus, the haptic rendering was based on a scaled version of the PD command rather than on the raw error alone.

In addition to varying the intensity, the *bHaptics* mapping also changed the number of active tactors as cue magnitude increased. At lower levels, the cue was localized to a smaller set of tactors, while higher levels expanded the stimulated area. This provided both a magnitude cue through vibration

intensity and an urgency cue through increased spatial extent of the stimulation. A similar haptic feedback strategy was utilized in prior full-body haptic cueing studies for augmented pilot perception, where vibrotactile cue intensity and spatial activation patterns were used to communicate task-relevant guidance and state information to the pilot [1, 4, 5].

2.7.1 Formation Deformation Feedback

Formation deformation feedback was used to inform the pilot when the follower vehicles departed from their desired relative positions with respect to the leader. This cueing mode was separated into longitudinal and lateral components.

Longitudinal deformation feedback was active throughout the course and was mapped to the vest through the pitch channel. These cues communicated whether the follower was lagging behind or advancing too close in the longitudinal direction. A chest cue indicated that increased forward pitch was required, whereas a back cue indicated that forward motion should be reduced.

Lateral deformation feedback was mapped to the armbands through the roll channel and was used to indicate deformation of the swarm footprint during turning flight. Unlike the longitudinal cue, lateral deformation feedback was not active for the entire gate-to-gate segment. Instead, it governed the lateral cueing only outside the gate-approach region. Once the swarm entered the final gate-approach zone, lateral formation-deformation feedback was disabled and replaced by gate-avoidance cueing.

To prevent overcueing during nominal formation flight, a lateral no-cue band was defined around the desired follower offset. Let Y_{Wingman} denote the nominal lateral follower spacing and W_{safe} denote the safe-band width. Then lateral formation-deformation cues were generated only when the follower

lateral offset error exceeded

$$|y_{\text{rel}} - Y_{\text{Wingman}}| > \frac{W_{\text{safe}}}{2}, \tag{2.7}$$

with no cue applied inside that deadband. In the implemented course, $W_{\text{safe}} = 2R_{\text{quad}}$, so the half-width of the no-cue region was equal to one quadcopter radius. A visual representation of this lateral deformation geometry is shown in Fig. 2.12.

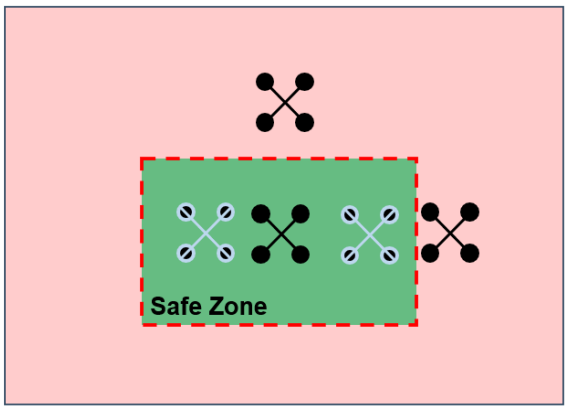


Figure 2.12: Illustration of the lateral formation-deformation cueing geometry. Lateral deformation cues are generated only when the follower departs from the lateral no-cue band about the nominal formation offset.

2.7.2 Gate-Avoidance Feedback

Gate-avoidance feedback governed the lateral cueing during the final approach to each gate. In the implemented logic, this mode became active when the swarm entered a gate-approach region extending a prescribed longitudinal distance upstream of the next gate. That activation threshold was defined as

$$e_{x,g} \leq X_{\text{th}}, \tag{2.8}$$

where $e_{x,g}$ is the remaining longitudinal distance to the next gate and X_{th} is the gate-avoidance activation distance. In the present implementation, $X_{\text{th}} = 49.5$ ft, corresponding to approximately

26% of the nominal gate-to-gate spacing.

Within this gate-approach region, cues were generated from the swarm footprint relative to a curved gate corridor rather than from formation deformation alone. The gate boundaries were widened upstream of the gate and contracted smoothly toward the physical gate opening, resulting in a parabolic funnel-shaped safe region. A representative form of that boundary is

$$y_{\text{bnd}} = y_{\text{gate}} \pm Y_{\text{start}} \left(\frac{e_{x,g}}{X_{\text{th}}} \right)^2, \tag{2.9}$$

where y_{gate} denotes the nominal gate edge and Y_{start} sets the upstream widening of the funnel. The resulting geometry is illustrated in Fig. 2.13.

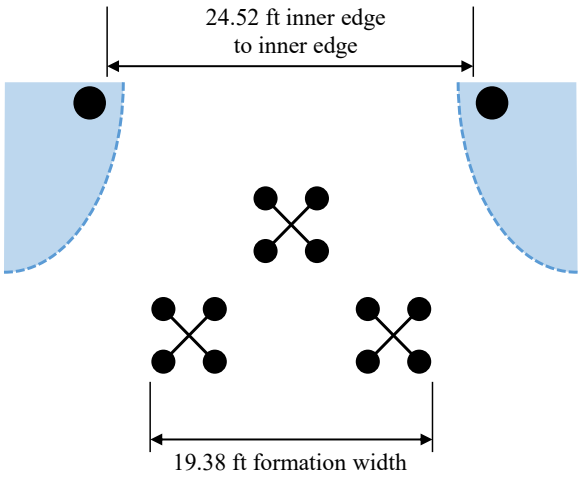


Figure 2.13: Curved gate boundary used during the gate-avoidance phase. The allowable lateral corridor contracts as the swarm approaches the gate.

Importantly, entering the gate-approach region did not by itself produce a haptic cue. Instead, gate-avoidance feedback was activated only when the swarm footprint violated the corresponding curved boundary. Thus, the gate-avoidance logic combined a longitudinal activation threshold with a lateral boundary-violation condition.

A left-arm cue indicated that the swarm was displaced to the right and required a corrective left-roll input, whereas a right-arm cue indicated the opposite. After the swarm cleared the gate, gate-avoidance feedback was disabled and lateral formation-deformation cueing resumed for the next gate-to-gate segment.

2.7.3 Cue Rendering and Tactor Activation

The *bHaptics* rendering logic used both vibration intensity and stimulated area to encode cue magnitude. For the roll channel, lower-magnitude cues activated a smaller pair of tactors on the appropriate arm, while larger-magnitude cues saturated that initial pair and expanded activation to additional tactors on the same arm. The same general strategy was used for the pitch channel on the vest. As a result, cue growth was represented not only by stronger vibration but also by a larger stimulated region on the body.

This behavior can be interpreted as a two-stage haptic encoding. For relatively small values of a_i, the cue magnitude was represented primarily through actuator intensity. Once the initial tactors reached their maximum command level, additional tactors were recruited to represent further growth in the cue command. Thus, higher cue levels did not reset the originally active tactors; rather, those tactors remained saturated while additional tactors were activated to represent the excess command magnitude. This provided a more salient indication of large errors without changing the directional interpretation of the cue.

2.7.4 Cueing Examples

Representative examples of the primary cueing cases used in the experiment are shown in Figs. 2.14–2.16. These examples illustrate how the haptic logic was applied to different swarm states during the course.

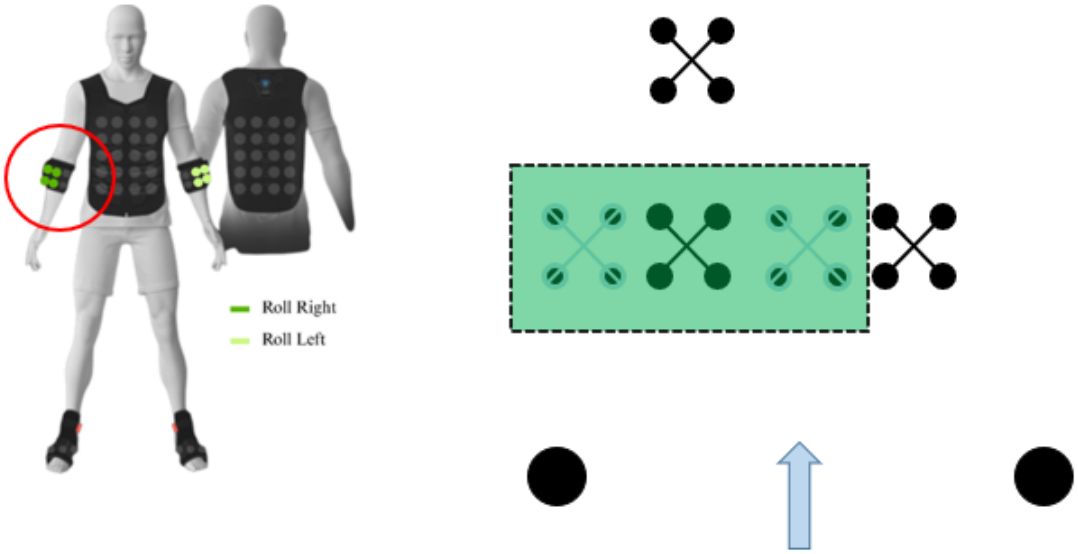


Figure 2.14: Example of lateral deformation cueing.

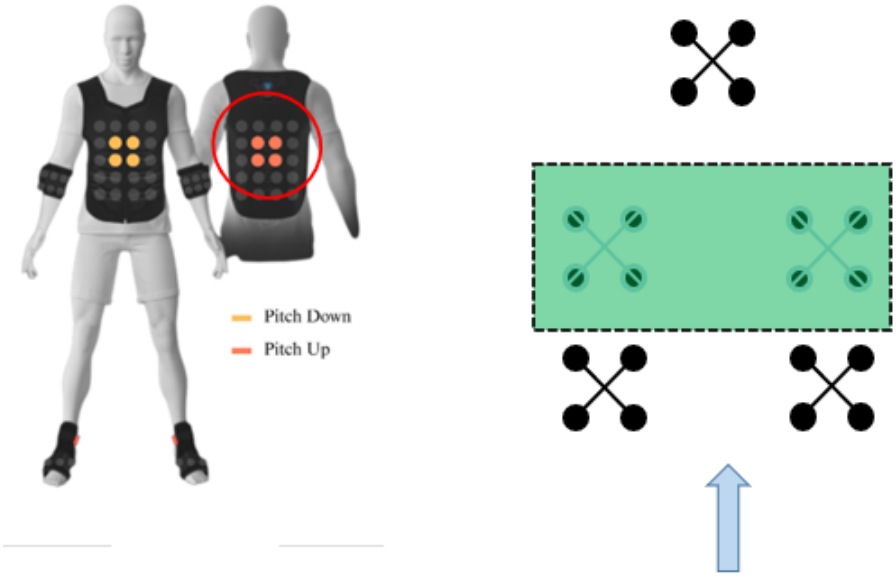
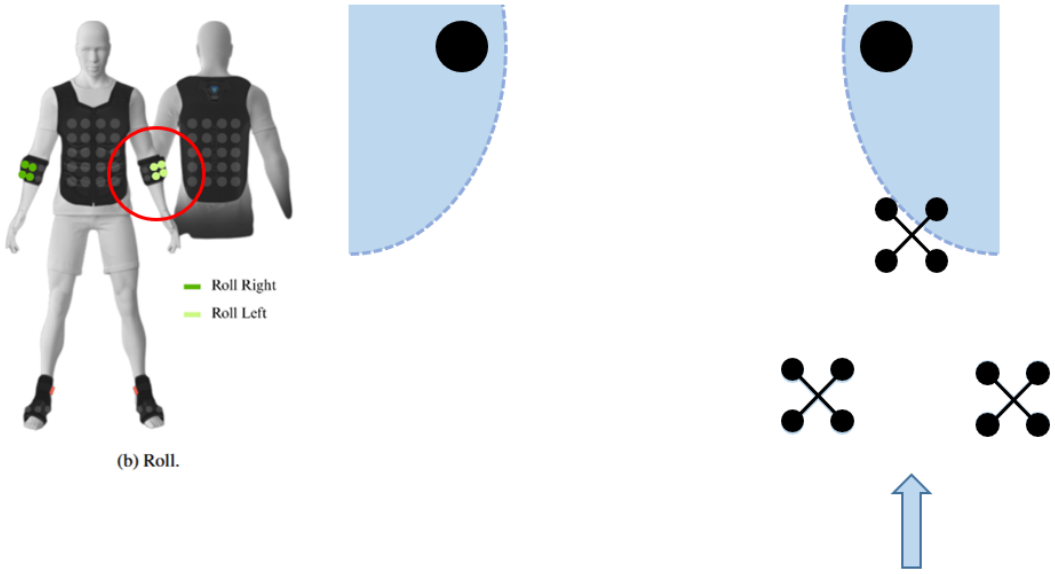


Figure 2.15: Example of longitudinal deformation cueing.



(b) Roll.

Figure 2.16: Example of gate-avoidance cueing.

2.8 Vehicle Modeling and Scaling Methodology

To establish dynamic equivalence between the simulated quadcopter and the full-scale rotorcraft Mission Task Elements (MTEs) defined in ADS-33, single-vehicle operating conditions were derived using Froude similarity. Froude scaling preserves the ratio of inertial to gravitational forces and is therefore well suited for translating maneuvering requirements and handling-qualities metrics across vehicles of different size.

The governing similarity parameter is the characteristic length ratio,

$$\lambda = \frac{L_{\text{UAV}}}{L_{\text{full}}}, \tag{2.10}$$

where L_{UAV} and L_{full} denote representative characteristic lengths of the quadcopter and the reference full-scale aircraft, respectively. The characteristic lengths used in this study are illustrated in Fig. 2.17.

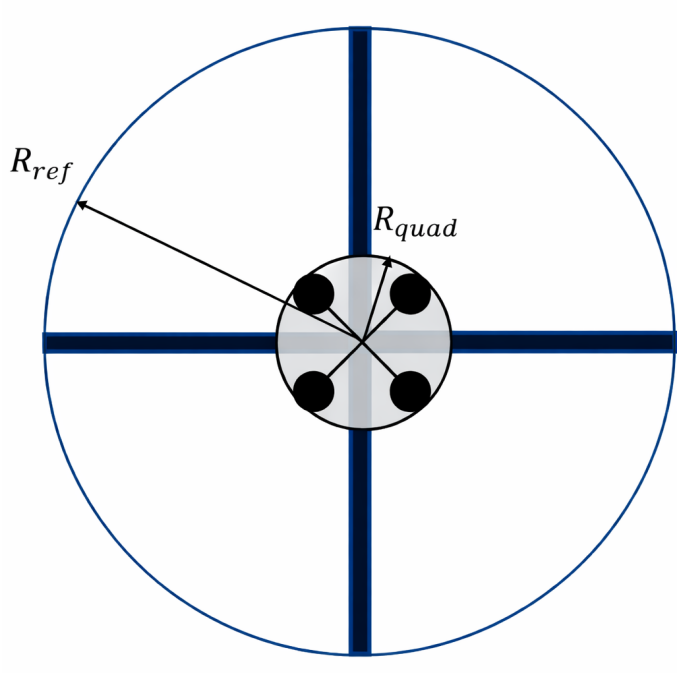


Figure 2.17: Characteristic lengths used for Froude scaling.

Under Froude similarity, kinematic quantities scale according to

$$V_{\text{UAV}} = \sqrt{\lambda}\, V_{\text{full}}, \tag{2.11}$$

$$T_{\text{UAV}} = \frac{T_{\text{full}}}{\sqrt{\lambda}}, \tag{2.12}$$

$$\omega_{\text{UAV}} = \frac{\omega_{\text{full}}}{\sqrt{\lambda}}, \tag{2.13}$$

such that dynamic responses remain consistent when expressed in nondimensional form. More generally, a physical quantity P with dimensional form $L^m T^n$ scales as

$$P_{\text{UAV}} = \lambda^m \left(\sqrt{\lambda}\right)^{-n} P_{\text{full}}. \tag{2.14}$$

The UH-60 Black Hawk was selected as the full-scale reference aircraft due to its central role in the development and validation of ADS-33 MTE criteria. The UH-60 main rotor diameter (53.8 ft) was used as the representative full-scale length. For the simulated quadcopter, the characteristic length was defined as the tip-to-tip rotor span (3.77 ft), resulting in a length ratio of

$$\lambda = \frac{3.77}{53.67}. \tag{2.15}$$

The full-scale velocity reference was taken as the UH-60 maximum speed of 150 kts. Applying Froude scaling yields the dynamically equivalent maximum quadcopter speed,

$$V_{\text{quad,max}} = V_{\text{H-60}} \sqrt{\frac{D_{\text{quad}}}{D_{\text{H-60}}}} \approx 40 \text{ kts}. \tag{2.16}$$

This single-vehicle scaling process was used to establish the target operating conditions for the

piloted leader vehicle, including the desired and adequate test speeds, as well as nominal task altitude used throughout the simulation study. These quantities are summarized in Table 2.1. The operating conditions in this table are distinct from the slalom-course geometry, which was scaled separately using swarm-level dimensions, as described in Section 2.10.

Table 2.1: Froude-scaled target operating conditions for the simulated quadcopter.

Parameter	Full-scale value	Scaled / adopted value
GVE desired speed	60.00 kts	14.72 kts
GVE adequate speed	40.00 kts	9.81 kts
DVE desired speed	30.00 kts	7.36 kts
DVE adequate speed	15.00 kts	3.68 kts
Altitude	30.48 m	1.83 m

The quadcopter model was subsequently trimmed and linearized at airspeeds ranging from hover to 40 kts in 5-knot increments. Owing to vehicle symmetry, the trimmed roll and yaw attitudes remained approximately zero across the flight envelope. The trim pitch angle decreased monotonically with increasing forward speed in order to generate the propulsive thrust required to overcome aerodynamic drag. The resulting trim states and control inputs were then used to support the stitched residualized model employed in the real-time simulation environment.

A representative view of the quadcopter model geometry generated from the vehicle dynamics model is shown in Fig. 2.18.

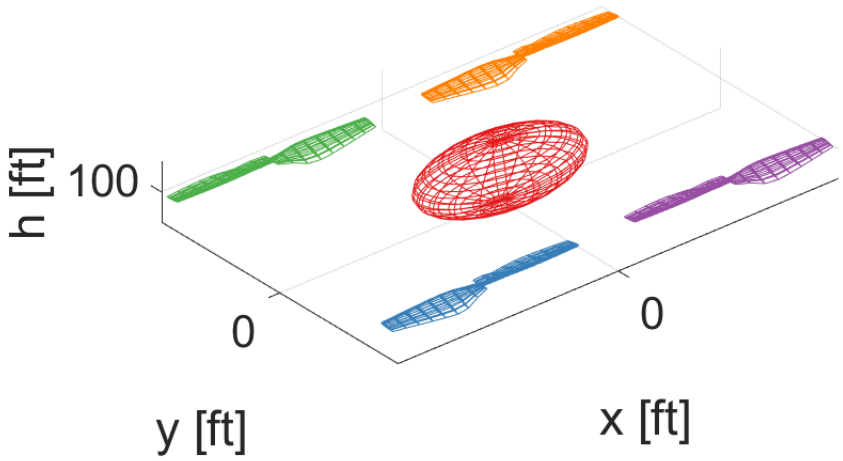


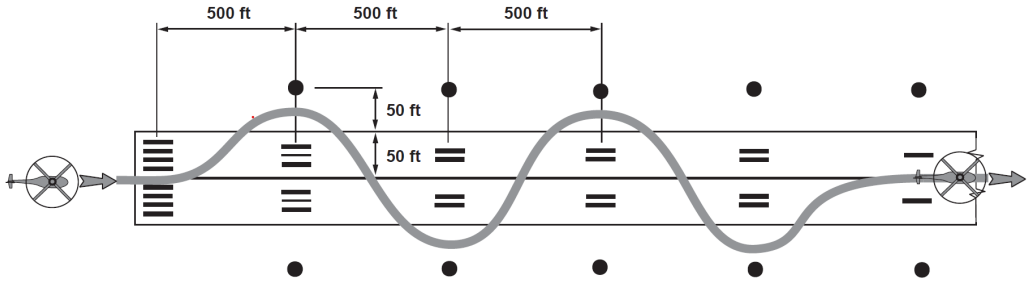
Figure 2.18: Quadcopter model geometry from the vehicle dynamics model.

2.9 Mission Task Element Design

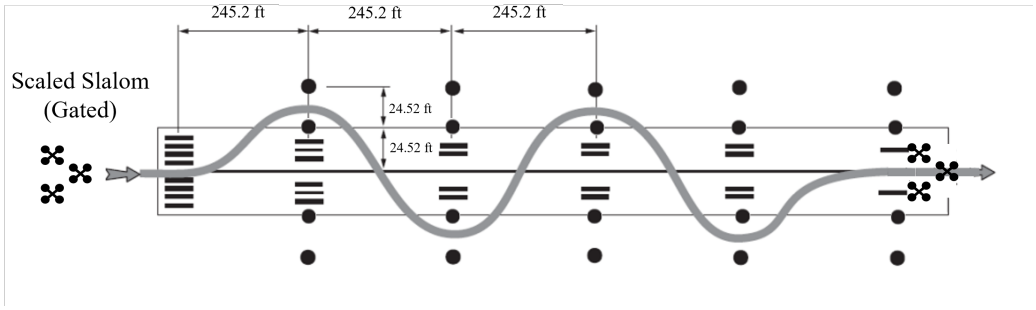
The Mission Task Element selected for this study was a gated slalom course. The objective of the task was to maneuver the swarm through a sequence of laterally offset gates while maintaining formation cohesion and avoiding pylon collisions. The slalom configuration was selected because it simultaneously challenges lateral control, longitudinal speed regulation, and formation maintenance, making it well-suited for evaluating the effectiveness of haptic feedback in a human–swarm control context.

The course consisted of one starting gate, four or five intermediate gates, and one end gate. Pre-test trials were conducted using four intermediate gates, while experimental trials used five. The experiment phases associated with these course configurations are outlined later in Section 2.11. Each intermediate gate was defined by two vertical pylons. Pylon colors indicated the required path through

the course, with red and gray patterns alternating between gates. For consistency across trials, pilots were instructed to always follow the gray gates which started on the left.



(a) Original Slalom.



(b) Scaled Slalom.

Figure 2.19: Original and scaled slalom MTE (Ref. 6).

The starting gate was represented by dashed runway markings rather than physical pylons. Pilots were briefed on the required maneuver for this gate prior to each trial to ensure consistent task execution.

2.10 Course Geometry and Parameterization

While the target operating conditions were derived from the characteristic dimensions of a single quadcopter, the slalom-course geometry was scaled separately using dimensions representative of the swarm as a whole. This distinction was necessary because the course was not designed for a single vehicle to pass through in isolation, but rather for the three-agent formation to maneuver through the

gates while preserving the desired relative geometry.

Accordingly, dimensions directly related to the course layout—including gate width, inter-gate spacing, and lateral gate offset from the runway centerline—were based on swarm-level characteristic distances rather than on single-vehicle dimensions alone. In particular, the governing reference quantities were taken from the overall formation footprint and the center-of-gravity to center-of-gravity spacing across the swarm. This approach ensured that the scaled course reflected the maneuvering requirements of the formation as a coordinated system rather than those of the leader alone.

The resulting geometry was derived using an abridged Froude-scaling procedure applied to the swarm reference dimensions. In this context, the scaling was used to preserve the relative spatial difficulty of the original ADS-33 slalom task while accounting for the finite width and turning behavior of the full formation. As a result, the scaled gate widths and gate spacings were larger than would have been obtained using only a single-vehicle characteristic length.

An additional safety factor was applied to the swarm-based geometric dimensions to ensure that the task remained feasible in real-time piloted simulation and to avoid making the course unrealistically restrictive for a three-agent formation. This adjustment was particularly important for gate width, since the swarm had to accommodate both the nominal formation footprint and the transient deformation that occurred during turning maneuvers.

Because the haptic logic operated on the swarm footprint rather than on the leader vehicle alone, the same swarm-level dimensions also informed the cueing boundaries used during the task. In particular, the lateral no-cue region for formation deformation and the gate-avoidance funnel were both defined relative to the formation geometry and the effective swarm envelope. This ensured that the haptic cues reflected the spatial constraints experienced by the swarm as a coordinated system rather than by a single vehicle in isolation.

The principal swarm-level geometric quantities derived from the implemented formation and cueing parameters are summarized in Table 2.2. These values were used in the slalom layout, formation definition, and haptic-boundary construction.

Table 2.2: Swarm-level geometric quantities used in the scaled slalom course and haptic logic.

Parameter	Value
Quadcopter radius, R_{quad}	1.61 ft
Nominal longitudinal follower offset, X_{Wingman}	8.07 ft
Nominal lateral follower offset, Y_{Wingman}	8.07 ft
Lateral safe-band width, W_{safe}	3.23 ft
Gate-avoidance activation distance, X_{th}	49.5 ft
Nominal gate-to-gate spacing	188.6 ft
Inner gate width	24.52 ft

The final course geometry used in the simulation therefore reflected two separate scaling processes: vehicle-based Froude scaling for the piloted leader's operating conditions and swarm-based scaling for the slalom-course layout and haptic-boundary definitions. This separation allowed the vehicle dynamics, swarm geometry, and task environment to remain mutually consistent while still respecting the geometric demands of coordinated formation flight.

2.11 Experimental Breakdown and Design

The simulation experiment was structured to evaluate pilot performance across four visual-and-haptic conditions while also giving participants time to become familiar with the control interface, virtual environment, and cueing strategy. The overall test setting and procedure are illustrated in Fig. 2.20. As shown in the figure, the experiment used a VR-based Prepar3D piloting setup and included both good visual environment (GVE) and degraded visual environment (DVE) cases, followed by a staged testing sequence consisting of setup, training, pre-test, four test modalities, and debriefing.



Figure 2.20: Overall simulation test setting and procedure. The figure shows the VR-based piloting setup used with Prepar3D, representative GVE and DVE pilot views, and the sequence of experiment phases from setup through debriefing.

Four modalities were pre-selected for inclusion in the final experiment. These included GVE without haptics, GVE with haptics, DVE without haptics, and DVE with haptics, referred to as modalities 1, 2, 3, and 4, respectively. Each subject participated in all four modalities, but the order in which they were presented was determined across four different groups. Each group began with a different modality, and the full sequence was assigned using a Latin square counterbalancing scheme, which helped mitigate order effects [26].

There were five trials performed for each modality, and both trial length and total modality duration varied depending on the participant. Subjects who maintained a higher average velocity typically completed all five trials in approximately 10 minutes, whereas subjects who fell below the desired speed often required 14–15 minutes. During the experiment, subjects were informed in real

time whenever a gate was missed.

Experiment breakdown

1. **Briefing and setup (30 min)**
    - Short PowerPoint presentation explaining the components of the experiment
    - Subjects were assisted with putting on the haptic equipment
    - Short pre-test survey
2. **Practice phase (15 min)**
    - Subjects performed 1–2 trials of each modality to become comfortable with the experimental setup
3. **Pre-test (15 min)**
    - Subjects performed five trials on a smaller slalom course consisting of four intermediate gates
    - This phase was used as a baseline measure of performance
4. **Actual test (45–70 min)**
    - Subjects performed five trials in each modality on a slalom course consisting of five intermediate gates
    - A 30-second rest period was provided after each trial
    - A 3–5 minute break was provided after each modality to complete the scales described in Section 2.12
    - Breaks were used to reduce mental workload, allow subjects to rest, and provide time for restroom use if needed
    - Additional modality-specific feedback was collected when provided by the participant

5. **Cleanup and debrief**

    - Subjects were assisted with removing and returning all equipment
    - General feedback and notes for improvement were collected
    - All equipment was sanitized and prepared for the next subject

A total of 25 subjects participated over the two-month testing period. Of these, 3 were women, 8 were drone pilots, and 1 was a helicopter pilot. The first 9 subjects participated during the initial month and were used for preliminary testing of the experiment. Based on feedback from these 9 participants, several major changes were made to improve and refine the experimental procedure. The remaining 16 subjects were used in the final data analysis. Their average age was 24.19 ± 2.54 years.

The main goal of the task was to navigate the swarm through each set of pylons in the slalom course while maintaining formation and meeting the required speed criteria.

2.12 Performance Metrics

The primary objective of the experiment was to guide the swarm through the slalom course while preserving formation and meeting the prescribed speed requirements. Performance was therefore evaluated using a combination of objective flight metrics and subjective pilot ratings.

The principal objective metrics used in the experiment are summarized below.

- **Missed gates:** A missed gate was defined as any event in which a pylon was struck by any agent in the swarm, or any case in which the full three-vehicle formation failed to pass successfully through the gate opening.
- **Formation distortion:** Swarm formation distortion was evaluated from the deviation of the follower vehicles from their nominal relative positions with respect to the leader. Because the

haptic feedback strategy was designed primarily to support formation maintenance, this metric served as a central indicator of performance.

- **Distance to pylons at failure:** For failed trials, particularly during early participant runs, the distance from the swarm to the gate pylons at the time of failure was also examined. This provided additional insight into how the swarm deviated from the intended path during missed gates or collisions.
- **Speed compliance:** Both GVE and DVE conditions had prescribed target speeds derived from the scaled ADS-33 task. Trials in which the swarm fell below the acceptable speed threshold were considered unsuccessful, even if all gates were cleared.
- **Trial completion time:** Total course completion time was recorded as a secondary measure related to speed control and overall task efficiency.

In addition to these objective measures, subjective pilot ratings were collected after each modality. Mental workload was assessed using the Bedford Workload Scale, and situational awareness was assessed using the China Lake Situational Awareness Rating Scale. Together, these measures were used to evaluate both the objective and perceived effects of haptic cueing and visual condition.

2.13 Data Collection and Analysis

A short pre-experiment survey was administered to collect participant background information relevant to interpretation of the results. The survey recorded the following items:

- age,
- gender,
- dominant hand,

- prior drone-piloting experience,
- prior general drone-flight experience and platform type, and
- video-game experience.

During each simulation trial, the states of all three vehicles were logged and exported for post-processing. These logged data included leader and follower translational states, rotational states, inertial positions, state derivatives, and pilot control inputs. In addition, quantities directly related to haptic cue generation—including the longitudinal and lateral error terms and their corresponding rates—were recorded in the same dataset.

Because the *bHaptics* interface rendered cues from scaled and discretized command values, the logged haptic-related quantities were used primarily to reconstruct cue timing, cue direction, and relative cue magnitude during post-processing rather than to store every individual tactor command explicitly. This allowed the analysis to identify when deformation and gate-avoidance cues occurred, how strongly they were commanded, and how those cues related to task performance.

Post-processing of the simulation data was performed primarily in MATLAB. The logged time histories were used to compute the main objective performance measures, including missed-gate percentage, formation distortion, speed compliance, and trial completion time. Subjective measures, including Bedford workload ratings and China Lake situational awareness ratings, were then compared against the objective metrics in order to assess the overall effects of haptic cueing and visual condition.

Chapter 3: Physical Demonstration and Experimental Setup

3.1 Quadcopter Components

3.1.1 Quadcopter Core Components

The quadcopter platform was constructed using hardware from the Quad Zero 3DR kit. The central element of the vehicle was the Quad Zero carrier board (Fig. 3.1), which integrates four 13 A BLHeli_S electronic speed controllers (ESCs) supporting 2–4S battery configurations. The carrier board also includes onboard sensing capabilities, including a VL53L1X time-of-flight (ToF) sensor, a Hall-effect current sensor, and a voltage sensor for power monitoring. These sensors enabled real-time measurement of vehicle state and power consumption during flight operations.



Figure 3.1: Quad Zero carrier board (from Ref. 7).

Low-level stabilization, state estimation, and actuator control were performed using a Control Zero H7 OEM flight controller (Fig. 3.2). The controller is based on an STM32H743 microcontroller with a 480 MHz Arm® Cortex®-M7 core, providing sufficient computational capability for real-time multirotor attitude and rate control. Flight controller sensor information is provided in Section 3.1.4.

Eight DShot-capable PWM outputs were used for motor actuation, and onboard data logging was supported via an industrial microSD card. The controller is compatible with both ArduPilot and PX4

firmware; for the present work, PX4 was used to interface with the swarm control and experimental architecture.

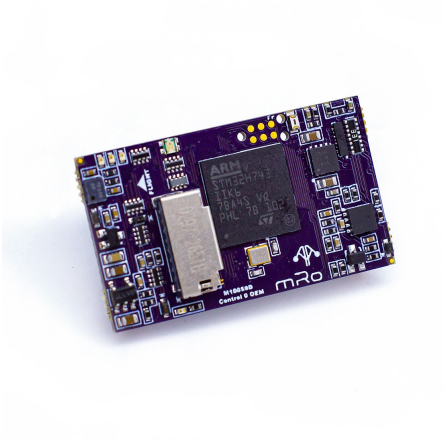


Figure 3.2: Control Zero H7 OEM flight controller (from Ref. 7).

3.1.2 Airframe Design and Mounting Hardware

The baseline vehicle utilized a stock 3DR Quad Zero frame constructed from fiber-reinforced plastic (FRP) with a 265 mm wheelbase. This frame provided a lightweight and sufficiently rigid platform for initial integration and testing, with a measured mass of approximately 21.85 g. The original design employed a three-screw motor mounting pattern and integrated battery retention features within the lower frame structure.

While the baseline frame enabled early system validation, several limitations were identified during testing. In particular, the three-point motor mounting configuration provided limited structural robustness under repeated loading, and the integrated battery clamp geometry restricted rapid battery replacement between experimental runs.



Figure 3.3: Baseline 3DR Quad Zero frame used during initial testing.

To address these limitations, a custom airframe was developed based on the original Quad Zero geometry while incorporating several key design modifications. The redesigned frame was manufactured using PETG-CF, providing improved durability and impact resistance for repeated indoor testing. The motor mounts were revised to use a fully circular geometry with four mounting holes, enabling improved load distribution and more secure motor attachment. In addition, the integrated battery clamps were removed in favor of a simplified central mounting region, allowing the battery to be secured using an external Velcro strap. This modification improved battery accessibility and reduced turnaround time between flights.



Figure 3.4: Custom PETG-CF frame with circular motor mounts and simplified battery mounting configuration.

Several PETG prototypes were initially fabricated using an in-lab Bambu X1C 3D printer to iterate on the design, with the final version produced in PETG-CF. The final airframe design reflects a balance between structural robustness, manufacturability, and ease of operation during experimental testing.

The GPS assembly was mounted above the primary frame using an elevated riser structure that positioned the GPS module away from the main airframe. This arrangement improved satellite visibility and helped reduce electromagnetic interference from the propulsion system and power electronics. The riser was retained in the updated vehicle configuration because it provided a compact, lightweight, and effective mounting solution for the GPS unit.



Figure 3.5: GPS mount and riser used to elevate the GPS module above the airframe.

3.1.3 Propulsion and Power Supply Components

The propulsion system utilized mRo M2-2202 brushless motors (9.1 g each) configured with a 12 mm mounting pattern and 4 mm motor shaft. When operated on a 2S battery configuration, this motor produced the thrust required for the scaled swarm maneuver while maintaining a compact form factor compatible with the Quad Zero airframe.



Figure 3.6: mRo M2-2202 brushless motor (from Ref. 7).

Each motor was paired with 7×5.5 in carbon fiber propellers selected to provide a balance between thrust generation and aerodynamic efficiency while minimizing rotational mass. This motor–propeller combination produced a measured static thrust of approximately 180 g per motor at maximum throttle when operated on a 2S battery configuration. The resulting thrust-to-weight capability provided

sufficient control authority for aggressive maneuvering within the scaled Mission Task Element envelope while maintaining efficient steady-state cruise performance.



Figure 3.7: 7×5.5 in carbon fiber propeller (from Ref. 7).

Electrical power was supplied by an mRo 2S 4200 mAh lithium-ion battery pack with a nominal voltage of 7.4 V and a continuous discharge rating of 3C. The battery mass was 126.8 g, representing a significant portion of the total vehicle weight and therefore an important contributor to overall thrust-to-weight ratio and flight endurance. Power was delivered through a BT2.0 connector with a JST-SH balance connector for cell monitoring and charging. The selected capacity provided sufficient flight time for repeated experimental trials while maintaining a compact form factor compatible with the quadrotor frame.



Figure 3.8: mRo 2S 4200 mAh lithium-ion battery pack (from Ref. 7).

3.1.4 Sensors and Receivers

The flight controller incorporates a triple-IMU architecture consisting of two Invensense/TDK inertial sensors (ICM-20602 and ICM-20948) and a vibration-isolated Bosch BMI088 unit, along with an internal magnetometer and a barometric pressure sensor. This redundant sensing configuration improves robustness to vibration-induced measurement noise and enhances state estimation reliability during aggressive maneuvers.

The quadrotor platform utilized a SAM GPS module based on the u-blox M8Q GNSS receiver integrated with an IST8308 magnetometer. The module supports multi-constellation satellite reception, providing robust global position estimates for outdoor flight operations. A manufacturer-recommended ground plane was incorporated to improve signal strength and reduce susceptibility to noise and multipath interference. The integrated magnetometer supplied heading information to the flight controller, supporting stable yaw estimation.



Figure 3.9: SAM GPS module with integrated IST8308 magnetometer (from Ref. 7).

The vehicle communication link was implemented using a 3DR DualBand Wi-Fi radio based on the u-blox ODIN-W260 module. The radio interfaces with the flight controller via a 6-pin JST-SH connector and supports both UDP and TCP communication. For the present work, the module was configured to transmit MAVLink messages between the vehicle and the ground control or experimental processing system. The radio can operate in either Station or Access Point mode, allowing flexible network configuration depending on the experimental setup. This wireless link enabled real-time telemetry streaming and command transmission during swarm testing.



Figure 3.10: 3DR DualBand Wi-Fi radio based on the u-blox ODIN-W260 module (from Ref. 7).

The radio control link was implemented using a FrSky R-XSR 2.4 GHz receiver operating under the D16 protocol. The receiver outputs RC channel data via SBUS to the flight controller, enabling

pilot stick inputs to be translated into commanded vehicle attitude and throttle responses during flight testing. Its compact, lightweight design facilitated integration into the quadrotor without significantly affecting mass distribution.



Figure 3.11: FrSky R-XSR radio receiver (from Ref. 8).

3.2 Quadcopter Construction

The quadcopters were assembled over several days using primarily the components provided in the 3DR Quad Zero kits. Additional materials included vibration-damping foam, wiring to connect the carrier board to the GPS and WiFi modules, and standard fasteners required for assembly. A representative fully assembled vehicle is shown in Fig. 3.12.

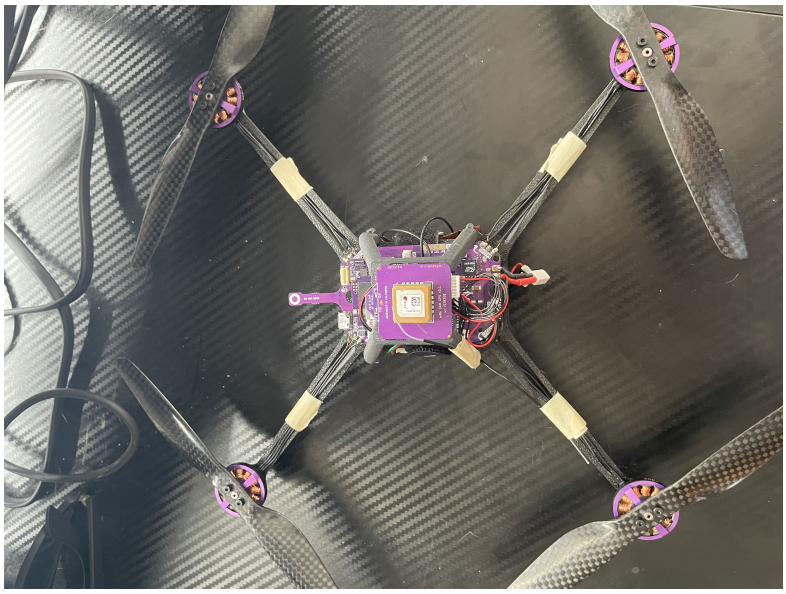


Figure 3.12: Fully assembled quadcopter used for the physical swarm demonstration.

Several components required soldered connections, including the motor leads to the carrier board, the FrSky receiver, and the main power leads. The carrier board was mounted to the frame using a combination of vibration-isolating foam and double-sided adhesive tape to reduce the transmission of motor-induced vibrations to the flight controller.

Assembly was performed in multiple phases. The initial phase consisted of mounting the motors to the frame using screws, soldering the FrSky receiver and power leads to the carrier board, and securing the carrier board to the frame. The second phase involved soldering the motor wires to the carrier board and preparing the GPS and WiFi modules, which were mounted to their respective structures using double-sided tape before being attached to the frame. The final phase included routing and connecting all remaining wiring between components, mounting the FrSky receiver to the frame, and organizing motor wiring along the arms using tape to ensure secure placement and minimize interference with the propellers.

Care was taken throughout the assembly process to ensure clean wiring, secure mechanical

connections, and adequate vibration isolation, all of which are critical for reliable flight performance and state estimation.

3.3 Flight Stack and Controller

PX4 version 1.17.0 beta was utilized as the primary flight stack for this experiment. Each drone relied on the pre-configured PX4 control laws for low-level stabilization and individual vehicle controllability. QGroundControl was used as the ground station software for parameter configuration, system monitoring, and basic testing throughout all phases of the experiment.

3.4 Pilot Input

Pilot inputs for the physical demonstration were provided using a Jumper T-Lite radio transmitter. The transmitter was used by the human operator to manually command the leader vehicle during flight testing. Through this interface, the operator provided the primary maneuvering inputs used to guide the swarm through the experimental environment, while higher-level swarm behavior was achieved through the leader–follower control architecture.

For the present implementation, the transmitter was connected only to the leader vehicle. The follower drones were not flown manually and did not receive direct stick commands from the operator. Instead, they responded indirectly to the pilot's actions through the swarm controller, which generated follower setpoints from the measured leader state.



Figure 3.13: Jumper T-Lite radio transmitter used to provide manual inputs to the leader vehicle (from Ref. 9).

3.5 Flight Modes

Different PX4 flight modes were used for the leader and follower vehicles in order to support the leader–follower swarm architecture. The leader vehicle was flown in PX4 Position mode, allowing the human operator to command translational motion through the radio transmitter while benefiting from onboard position stabilization. This provided a stable and repeatable reference vehicle for the remainder of the swarm.

The follower vehicles, by contrast, were operated in PX4 OFFBOARD mode. In this mode, the followers did not rely on direct pilot stick inputs for trajectory generation. Instead, they received externally generated position setpoints from the ground-station computer through the swarm-control framework. These setpoints were updated continuously during flight so that the followers could

maintain their prescribed relative positions with respect to the leader.

The use of Position mode for the leader and OFFBOARD mode for the followers reflected the distinct functional roles of the vehicles within the swarm. The leader served as the manually piloted reference vehicle, whereas the followers served as autonomously commanded agents whose motion was determined by the centralized control architecture.

3.6 High-Level Communication and Control Architecture

The leader vehicle served as the manually piloted reference aircraft, while each follower vehicle was commanded autonomously through PX4 OFFBOARD mode.

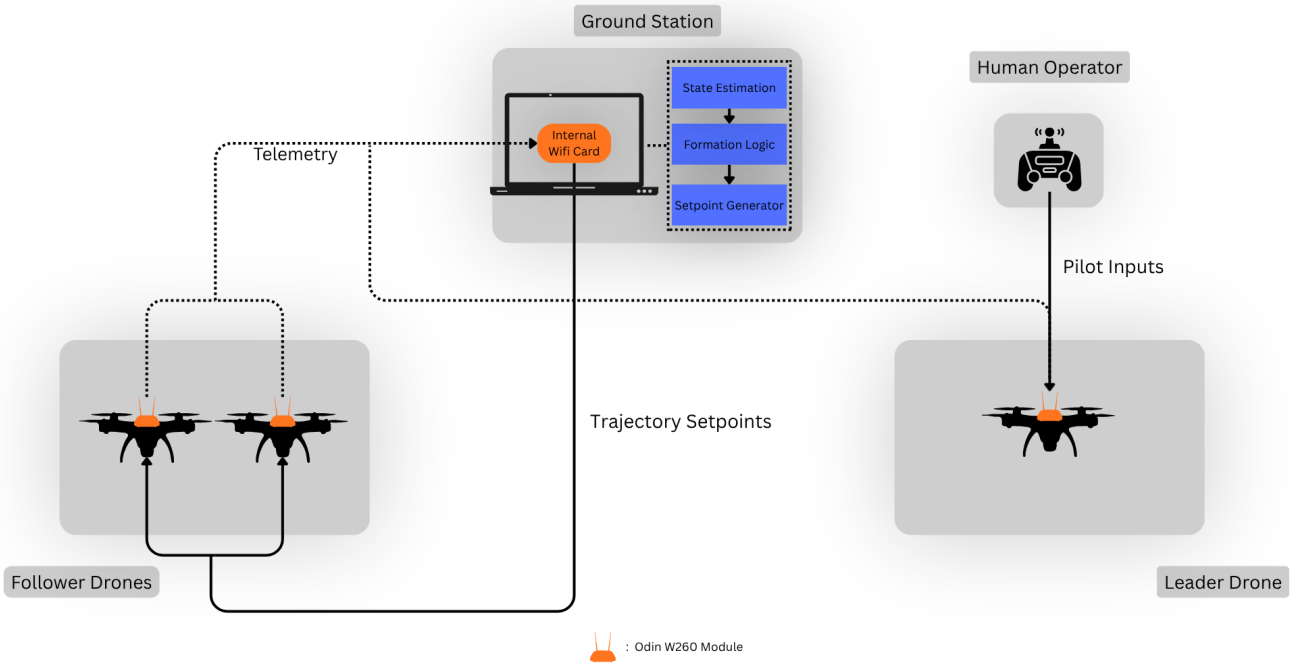


Figure 3.14: High-level communication and control architecture

Communication between the vehicles and the ground station was provided through ODIN-W260 radio modules configured in station mode. These modules enabled bidirectional telemetry and

command exchange over the local wireless network. Vehicle telemetry, particularly the leader state estimate, was transmitted from the aircraft to the computer, where the swarm controller computed the desired follower trajectories. As shown in Fig. 3.14, the resulting follower commands were then sent back to the corresponding vehicles as OFFBOARD setpoints. In this way, the laptop acted as the central node for both state collection and command generation.

A centralized control architecture was selected for this experiment primarily for simplicity and hardware practicality. The small quadcopters used in the physical demonstration were not equipped with dedicated companion computers or other onboard processors capable of handling higher-level swarm logic. As a result, implementing the formation controller on the ground station was the most direct and reliable approach.

At each control-loop iteration, telemetry from the leader and followers was processed on the ground-station computer. The controller used these data to estimate the current swarm geometry and compute the desired follower positions relative to the leader. These positions were based on predefined formation offsets in the leader frame. In the broader experimental context, the same state information could also be compared against prescribed slalom gate locations and desired vehicle trajectories in order to evaluate path-following performance, formation maintenance, and tracking error throughout the task.

Although this centralized method introduced a dependence on the ground-station communication link, it was sufficient for the scale and objectives of the present demonstration. The approach provided a practical means of implementing real-time leader–follower swarm control without requiring direct inter-agent communication or distributed onboard planning.

3.7 Port and Server Flow

To ensure that telemetry and commands were associated with the correct vehicle, each aircraft was assigned a distinct MAVLink communication path through the ground-station computer. This separation was especially important because multiple vehicles were operated simultaneously on the same local network. Without clear port separation, there would be a risk of duplicate telemetry binding, ambiguous vehicle identification, or commands being sent to the wrong aircraft.

Each ODIN-W260 link delivered MAVLink traffic from its associated vehicle to a dedicated UDP input port on the ground station. For the physical three-vehicle configuration used in this work, the leader, follower 1, and follower 2 were assigned UDP input ports 14550, 14551, and 14552, respectively. These input streams were then managed through separate MAVProxy instances. MAVProxy served as a routing and forwarding layer, allowing each vehicle's telemetry stream to be relayed to multiple local destinations without modifying the aircraft-side configuration.

For each vehicle, MAVProxy forwarded the incoming telemetry stream to two local UDP output ports. One output path was directed to a dedicated `mavsdk_server` instance, which provided the software interface used by the Python-based swarm controller. The second output path was reserved for QGroundControl, allowing simultaneous monitoring, visualization, and debugging during operation. This port-level separation provided a convenient way to distribute the same telemetry stream to multiple software components while preserving a clean one-vehicle-per-port structure.

Figure 3.15 illustrates the telemetry-routing and software-interface chain used for each individual vehicle within the swarm. Although Fig. 3.15 is presented as a left-to-right routing sequence for clarity, communication through the MAVLink and gRPC interfaces was bidirectional during operation, supporting both telemetry reception and command transmission.

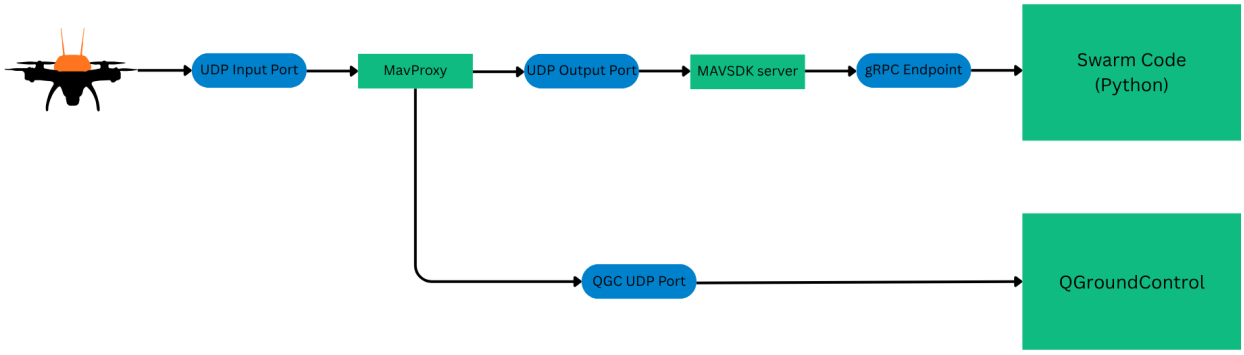


Figure 3.15: Per-vehicle MAVLink routing and software-interface flow used in the physical swarm architecture. Each vehicle transmitted telemetry to a dedicated UDP input port on the ground-station computer. MAVProxy then forwarded that stream to a vehicle-specific UDP output for `mavsdk_server` and to a separate UDP output for QGroundControl. The Python swarm code interfaced with each vehicle through the corresponding gRPC endpoint exposed by `mavsdk_server`.

In the Python control architecture, `mavsdk_server` acted as the interface layer between the raw MAVLink telemetry stream and the higher-level MAVSDK Python API. A separate `mavsdk_server` process was launched for each vehicle so that each drone maintained an isolated software connection. Each server listened to a vehicle-specific UDP output from MAVProxy and exposed a corresponding local gRPC endpoint to the swarm-control code.

Table 3.1 summarizes the specific port assignments used in the three-vehicle configuration. This structure maintained a one-to-one mapping between each vehicle, its telemetry stream, and its associated software interface. In practice, this reduced ambiguity during startup and simplified debugging by ensuring that the leader and each follower could be monitored and addressed independently.

Table 3.1: Port assignments used for leader and follower vehicles in the physical swarm architecture.

Function	**Leader**	**Follower 1**	**Follower 2**
UDP input port from ODIN-W260	14550	14551	14552
MAVProxy output to `mavsdk_server`	14600	14601	14602
MAVProxy output to QGroundControl	14700	14701	14702
gRPC endpoint port for Python interface	50051	50052	50053

Using a dedicated `mavsdk_server` instance for each vehicle improved the reliability of multi-vehicle operation in Python. In particular, it reduced the likelihood of duplicate telemetry association, where two logical software connections might inadvertently bind to the same physical drone. Startup helper scripts were therefore used to launch the server processes, verify port availability, and maintain the intended one-to-one mapping between each drone, its telemetry stream, and its software interface.

This communication structure allowed the leader and follower vehicles to be addressed independently while still operating within a unified swarm-control framework. It also made the system easier to debug, since telemetry flow could be checked at each stage of the chain: radio link, UDP input, MAVProxy forwarding, `mavsdk_server` connection, gRPC interface, and finally the Python control node.

3.8 Haptic Feedback Interface

The physical demonstration used the same bHaptics hardware as the simulation experiment, consisting of a vest for pitch-related cues and armbands for roll-related cues. The same general cue-mapping strategy was retained so that the physical and simulated haptic interfaces remained as consistent as possible.

As in simulation, longitudinal formation cues were intended to be rendered through the vest, while lateral deformation and gate-related cues were intended to be rendered through the armbands. The main difference between the two implementations is that the simulation used exact vehicle and gate states, whereas the physical system must generate cues from estimated drone positions and pre-recorded gate locations.

For the intended physical slalom implementation, gate pylon locations were to be recorded in latitude and longitude, converted into the coordinate frame used by the swarm controller, and then compared against the current estimated vehicle positions. In this way, gate-related haptic cues could be generated from the estimated position of the swarm relative to the physical course geometry.

At the time of writing, preliminary physical haptic testing had focused primarily on verifying that vehicle-state information could be routed through the software framework and rendered to the operator in real time. Initial validation was performed using attitude-based cueing. Full physical validation of position-based formation-deformation and gate-avoidance cueing had not yet been completed.

3.9 Software Architecture

The physical flight-test software was organized as a modular Python-based architecture in which a central wrapper script coordinated the major functions required for the experiment. These functions included vehicle communication, follower OFFBOARD control, swarm-command generation, haptic cue generation, configuration management, safety monitoring, and data logging. This modular structure simplified testing and debugging by separating high-level coordination from lower-level control and support functions. A description of the major software components is provided below.

- **Supervisory Control**
  - Integrated the major software modules required for the physical experiment and coordinated their execution at predefined update rates.
  - Facilitated communication among modules and ensured that the required data were saved for post-processing analysis.
- **Offboard Control**
  - Managed PX4 OFFBOARD mode for the follower vehicles and handled their transition into and out of swarm operation.
  - Continuously transmitted follower position setpoints to maintain the commanded trajectories relative to the leader.
- **Swarm Controller**
  - Computed the desired follower positions from the leader state and the prescribed formation geometry.
  - Compared measured follower states to their nominal relative positions and generated the corresponding command setpoints.
- **Haptic Interface**
  - Calculated the haptic cues associated with position or formation-tracking errors and determined their direction and magnitude.
  - Transmitted the corresponding signals to the bHaptics devices to provide the pilot with real-time tactile feedback.
- **Configuration**
  - Contained the high-level experimental settings and tunable parameters used throughout the test framework.

  - Allowed rapid adjustment of communication settings, formation offsets, control limits, cueing thresholds, and safety bounds without modifying the main control logic.

- **State Representation**
  - Defined the data structures used to represent vehicle states, swarm geometry, and command variables.
  - Provided a consistent way to organize quantities in common coordinate frames for control and haptic calculations.

- **Logging and Safety Monitoring**
  - Recorded telemetry and flight-test data for the participating vehicles for later analysis.
  - Monitored key flight conditions and enforced safety-related limits, allowing the experiment to halt or revert to a safer mode if required conditions were not met.

3.10 Pre-experiment Simulation and Testing

Prior to the physical experiments, JMAVSim was utilized to perform a series of preliminary tests of the experiment code. These tests primarily consisted of communication checks, validation of trajectory setpoint transmission, and general verification of swarm logic and experiment flow. Although the simulation environment did not replicate the full physical experiment setup, it provided a useful platform for confirming that the overall software pipeline functioned as intended before implementation on the physical vehicles.

Later, a VICON-equipped setup was used to verify the stability and handling qualities of the physical drones. The room featured 12 Vantage V8 cameras that were carefully placed and calibrated for proper use with the VICON Tracker software. A single reflective marker was mounted on top of

the quadcopter so that its position could be tracked by the camera system. Data from the VICON host computer was gathered through a direct network connection using an external Wi-Fi module. The UDP packets were collected and filtered using the VICON Python SDK, then translated into PX4-compatible position estimates. These estimates were transmitted back through the internal Wi-Fi network to the drone's ODIN-W260, where they were fed into the quadcopter's EKF2, allowing for more stable position hold.

This testing also revealed additional instabilities related to the drone's center of gravity and leveling. As a result, certain setup and calibration procedures were repeated in order to further improve stability prior to the main experimental trials.

3.11 Experimental Design

The physical portion of this work was intended as a proof-of-feasibility demonstration of the leader–follower swarm architecture and associated haptic framework rather than as a full human-subject experiment equivalent to the simulation study. The primary objective was to demonstrate that the centralized swarm controller, telemetry-routing structure, PX4 flight modes, and haptic communication pipeline could operate together on the real quadcopter platform.

In the intended test configuration, the leader vehicle was flown manually by a human operator using the Jumper T-Lite transmitter while the follower vehicles operated in PX4 OFFBOARD mode. Preliminary testing focused on reduced-complexity scenarios emphasizing stability, communication reliability, mode transitions, and basic follower tracking before extension toward gated slalom maneuvers. Because the physical campaign was still in progress at the time of writing, this section describes the implemented platform and intended demonstration methodology rather than a completed formal

experimental protocol.

3.12 Data Collection and Analysis

Data collection for the physical demonstration was centered on vehicle telemetry, pilot inputs, and software-status logging from the swarm-control framework. PX4 logs were used to record drone states and other onboard quantities made available through the flight stack, including position, velocity, attitude, flight mode, and related state-estimation variables. Pilot inputs to the leader vehicle were also logged for comparison with the measured vehicle response.

In addition, the ground-station framework recorded `mavsdk_server` status and connection information in order to monitor communication continuity and interface health during operation. Further logging was intended to include commanded haptic cues, including their timing, direction, and relative magnitude. For the preliminary physical tests, analysis was primarily diagnostic and focused on trajectory comparisons, follower tracking quality, communication reliability, mode transitions, and overall haptic-command routing.

3.13 Limitations of Methods

The physical implementation introduces several practical limitations relative to the simulation environment. In particular, vehicle states and gate locations are estimated rather than known exactly, resulting in uncertainty in both follower tracking and haptic cue generation. In addition, the centralized control architecture introduces communication latency and dependence on continuous telemetry between the vehicles and the ground station. At the time of writing, the physical testing campaign remained limited to preliminary demonstrations, and full validation of position-based haptic cueing in multi-

vehicle flight had not yet been completed. Together, these limitations define the practical context for the preliminary physical results presented in Chapter 4.

Chapter 4: Results

4.1 Simulation Results

4.1.1 Example of Slalom Performance

A representative simulation run is shown in Figure 4.1 for the GVE+H condition. The leader trajectory is plotted together with the two follower trajectories as the three-vehicle formation progresses through the slalom course. In this example, the full test course was used, consisting of six gates distributed over roughly 1230 ft of travel.

Also shown in the figure are the curved boundary regions associated with each gate. These regions define the allowable lateral space available to the swarm as it approaches and passes through each obstacle. Because the boundary contracts near the gate plane, the task becomes more restrictive as the formation closes in on the obstacle, requiring more accurate lateral positioning at the moment of passage.

The trajectories indicate that the pilot was able to preserve the overall shape of the formation while completing the maneuver. The follower paths remain close to the leader path for most of the task, although some separation is visible during the turning portions of the slalom. Even with these transient deviations, the vehicles remain well-behaved and pass through the gated course successfully in this trial.

This example illustrates the combined control demands of the task. The operator was required to guide the leader through the course while indirectly maintaining the placement of the followers, rather than simply flying a single vehicle through a visual path. In that sense, the figure provides a qualitative view of the coupled navigation and formation-maintenance problem that is quantified in the following subsections.

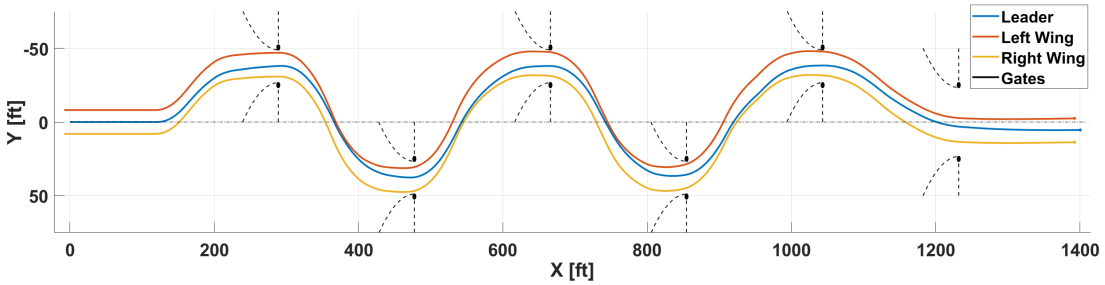


Figure 4.1: Representative piloted simulation trajectory in the GVE+H condition.

4.1.2 Formation-Keeping Performance

One of the main goals of the experiment was to determine whether haptic cueing improved the operator's ability to preserve the nominal swarm geometry during the slalom maneuver. To evaluate this, formation distortion was measured in both the longitudinal and lateral directions using the root-mean-square (RMS) value of the corresponding error histories. In general, smaller RMS values indicate that the followers remained closer to their desired relative positions throughout the run.

The RMS error for a generic signal $e(t)$ is defined as

$$e_{\mathrm{RMS}} = \sqrt{\frac{1}{T}\int_0^T e^2(t)\,dt} \tag{4.1}$$

where T denotes the trial duration. In this work, Eq. 4.1 was applied separately to the longitudinal deformation error e_x and the lateral deformation error e_y.

The resulting distributions are shown in Figure 4.2. In the longitudinal direction, the median RMS deformation decreased from approximately 0.25 ft in the Visual GVE condition to 0.11 ft in the GVE+H condition, corresponding to an improvement of roughly 56%. A similar trend was observed in DVE, where the median longitudinal RMS deformation decreased from approximately 0.18 ft to 0.09 ft when haptic cueing was added.

A similar but less uniform trend appears in the lateral direction. In GVE, the median lateral RMS deformation changed from approximately 0.12 ft in the visual-only condition to 0.10 ft with haptics, indicating a modest improvement. In DVE, the median lateral RMS deformation decreased from approximately 0.25 ft to 0.17 ft, suggesting that haptic cueing provided more noticeable lateral-formation support under degraded visibility.

Another important feature of Figure 4.2 is the spread of the data. In several of the haptic conditions, the interquartile range appears smaller than in the corresponding visual-only cases, indicating that performance was not only improved in magnitude but also made more repeatable across participants. From a swarm-control perspective, this is valuable because reduced variability implies more predictable human supervisory performance.

Taken together, these results support the conclusion that haptic feedback improved formation keeping in both axes, with the clearest benefits appearing in longitudinal deformation and in lateral deformation under degraded visual conditions.

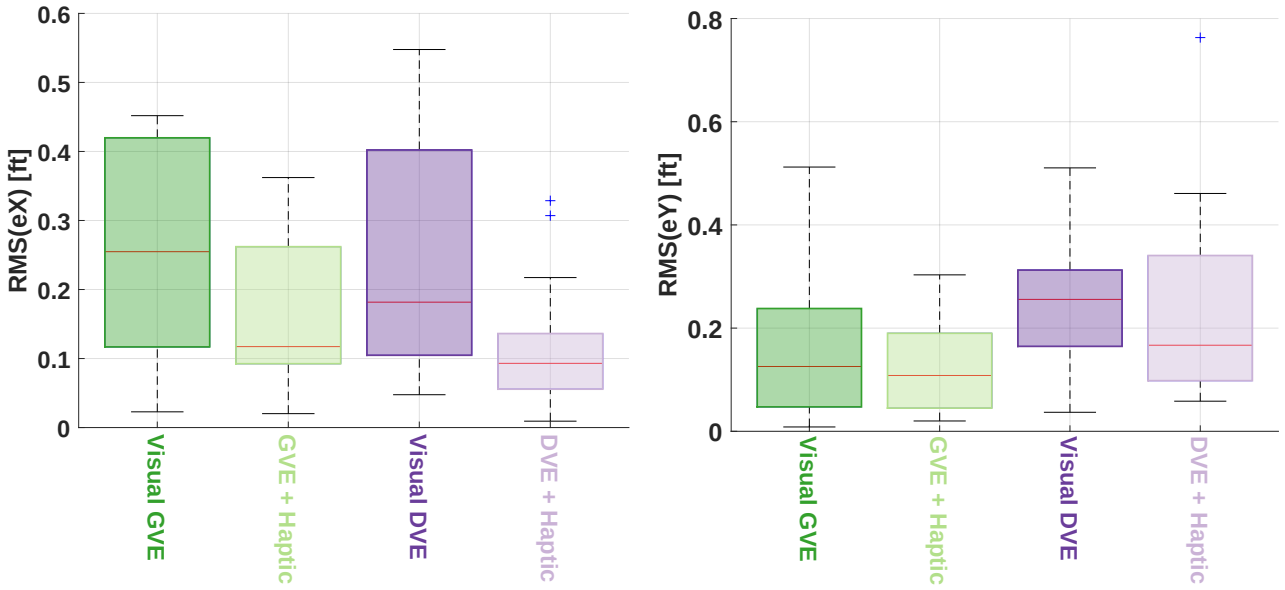
4.1.3 Gate Avoidance Performance

In addition to formation quality, task performance was also evaluated in terms of successful gate traversal. For this purpose, the percentage of missed gates was used as a simple objective indicator of obstacle-avoidance performance.

The percentage of missed gates was computed as

$$P_{\text{miss}} = \frac{N_{\text{miss}}}{N_{\text{tot}}} \times 100 \tag{4.2}$$

The results are shown in Figure 4.3. The median missed-gate percentage was approximately



(a) Longitudinal deformation.

(b) Lateral deformation.

Figure 4.2: Formation-keeping performance across all experimental conditions, shown using RMS deformation error. Lower values correspond to tighter maintenance of the nominal formation.

19% for Visual GVE and 13% for GVE+H, indicating a reduction of about 6 percentage points when haptic cueing was added in the good-visibility condition. By contrast, the DVE cases showed median missed-gate percentages of approximately 25% for both Visual DVE and DVE+H, indicating that the effect of haptics on gate-clearance performance was negligible under degraded visibility.

The strongest overall trend in Figure 4.3 is the difference between good-visibility and degraded-visibility conditions. Both DVE cases exhibit higher missed-gate percentages than their corresponding GVE cases, indicating that the visual environment had a stronger effect on obstacle avoidance than the addition of haptic cueing.

4.1.4 Mental Workload

Subjective workload was assessed using the Bedford Workload Scale. The full scale is provided in Appendix A.

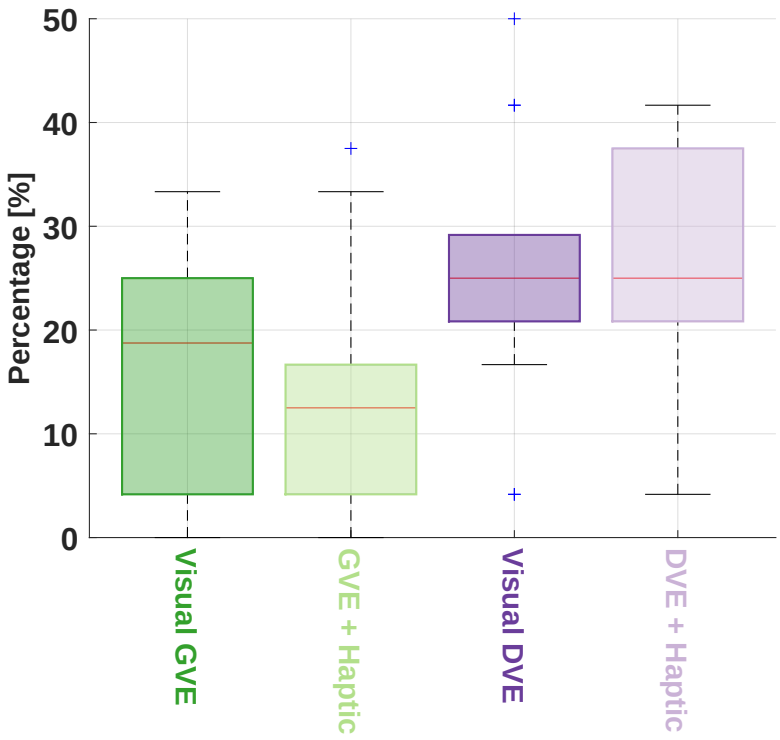


Figure 4.3: Percentage of missed gates across the four experimental conditions. Lower values indicate improved obstacle-avoidance performance.

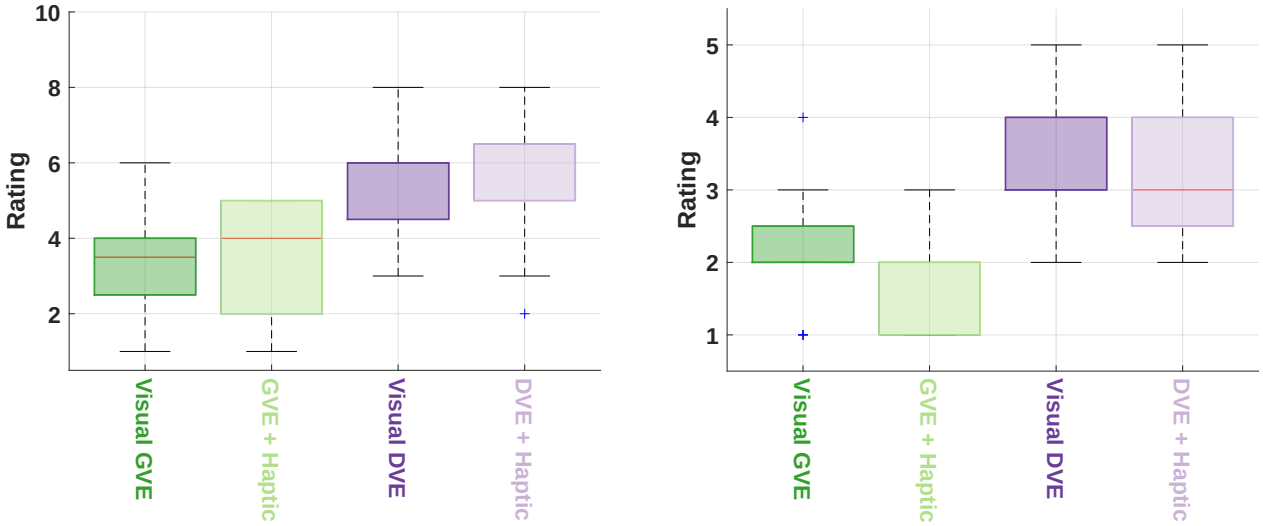
The workload ratings are shown in Figure 4.4(a). In the good-visibility conditions, the Bedford ratings were centered around 3.5–4. In the degraded-visibility conditions, the ratings shifted upward and were centered around 5–6, showing that DVE substantially increased perceived task difficulty.

Within each visual condition, the addition of haptics did not appear to cause a major change in workload. This suggests that participants were able to use the haptic information without perceiving it as a significant additional burden.

4.1.5 Situational Awareness

Situational awareness was evaluated using the China Lake scale. The full scale is provided in Appendix A.

The results are shown in Figure 4.4(b). In the good-visibility condition, the China Lake ratings were centered around 2 in both cases, with little visible difference between the visual-only and haptic conditions. In the degraded-visibility condition, the ratings shifted downward from approximately 4 in the visual-only case to approximately 3 with haptics, indicating improved perceived awareness.



(a) Bedford workload rating.

(b) China Lake situational awareness rating.

Figure 4.4: Subjective evaluation of workload and situational awareness across the four experimental conditions.

4.1.6 Participant Feedback

Qualitative comments collected from participants were generally consistent with the quantitative results and are summarized below.

Haptics

- Deformation haptics provided crucial information for maintaining swarm formation.
- Gate haptics were effective in guiding navigation toward the center of the gate.
- Haptic feedback occasionally increased task difficulty, but consistently improved pilot situational

awareness.

- Curved gate boundaries caused haptic cues to occur too late, often functioning as impact indicators rather than avoidance cues.
- The haptic feedback system occasionally presented excessive information, making it challenging for pilots to maintain consistent performance across all tasks.

Task Design

- The experiment's progression from practice to pre-test to trial was beneficial for gradually developing pilot proficiency.
- Coordinated turns would have allowed for easier task execution overall.
- The short distance between gates made it difficult to fully anticipate required maneuvers.
- Missing a gate felt overly penalizing at times, consuming a disproportionate amount of pilot workload and detracting from other tasks.

Visual Environment

- The DVE environment provided a challenging yet achievable experimental setting.
- In GVE, increased visual feedback reduced overall task workload, allowing pilots to maintain conversations during task execution.
- Increased visual feedback in GVE enabled a more relaxed task experience.
- The airspeed indicator displayed in the HUD enabled pilots to better estimate speed in DVE, where depth perception is significantly reduced.
- Limited visual feedback in DVE made it difficult to judge proper centering when transitioning between gate pylons.

Overall, the participant feedback reinforces the broader trends observed in the quantitative results. Haptics were viewed as especially useful for formation awareness and situational understanding, while the visual environment remained the dominant factor in perceived obstacle-navigation difficulty.

4.2 Physical Demonstration Results

The physical portion of this work was used to evaluate whether the leader–follower swarm architecture developed for simulation could be translated to a real PX4-based quadcopter platform. Unlike the simulation study, this portion was not completed as a full human-subject experiment or full slalom-course evaluation. Instead, the physical testing focused on implementation feasibility, including vehicle stability, position estimation, communication reliability, OFFBOARD command transmission, follower tracking behavior, and basic haptic-system integration.

Preliminary hover and position-hold tuning was first performed in the Maryland Aerial Robotics Lab. These tests were used to verify basic vehicle assembly, flight-controller configuration, sensor behavior, and hover stability before attempting multi-vehicle operation. This stage of testing was important because reliable low-level vehicle behavior was required before the swarm-control architecture could be evaluated. During these tests, vehicle setup and calibration were refined, and issues related to stability, position hold, and general handling were identified before moving to larger-scale flight testing.

After the initial tuning phase, additional testing was performed at the Fearless Flight Facility (F3). The larger indoor flight environment allowed the communication architecture, position-estimate routing, and leader–follower swarm functionality to be tested under more realistic operating conditions. The primary physical swarm configuration consisted of two vehicles: one manually piloted leader and

one OFFBOARD-controlled follower. The leader vehicle was flown in PX4 Position mode using the radio transmitter, while the follower vehicle received externally generated position setpoints from the ground-station swarm controller.

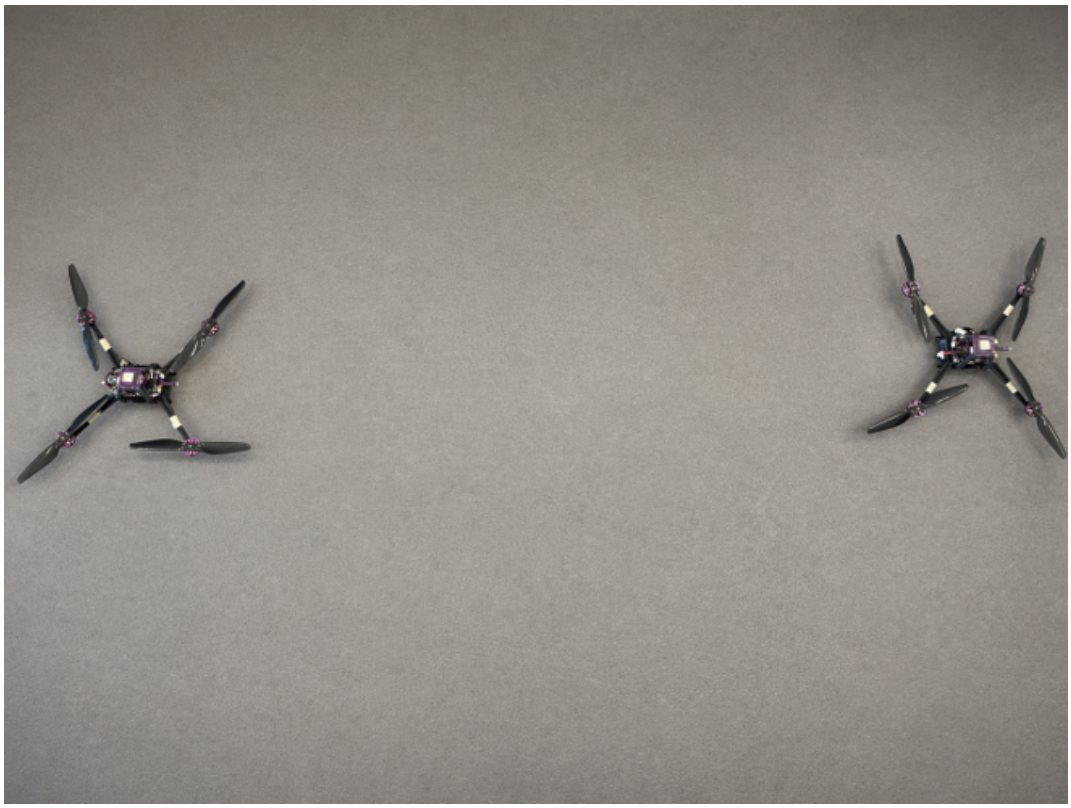


Figure 4.5: Two-vehicle physical test setup used for preliminary leader–follower swarm testing. The leader was manually piloted, while the follower was commanded through PX4 OFFBOARD mode.

During the F3 testing, the ground-station architecture successfully received telemetry from the vehicles, processed the leader state, computed the corresponding follower setpoint, and transmitted commands back to the follower. This demonstrated that the overall communication and control chain was functional in flight. In particular, the testing showed that the follower could respond to real-time leader-relative position commands rather than only to pre-scripted trajectories. This result was significant because it confirmed the basic feasibility of the centralized leader–follower approach on the custom quadcopter platform.

A representative physical trajectory from preliminary testing is shown in Fig. 4.6. In this example, the follower generally tracked the commanded motion relative to the leader, demonstrating that the swarm controller, MAVLink routing, mavsdk server interface, and PX4 OFFBOARD setpoint pipeline were able to operate together during flight. Some delay was observed between leader motion and follower response, which is expected for this architecture because leader state information must first be transmitted to the ground-station computer, processed by the swarm controller, and then sent back to the follower as an OFFBOARD command.

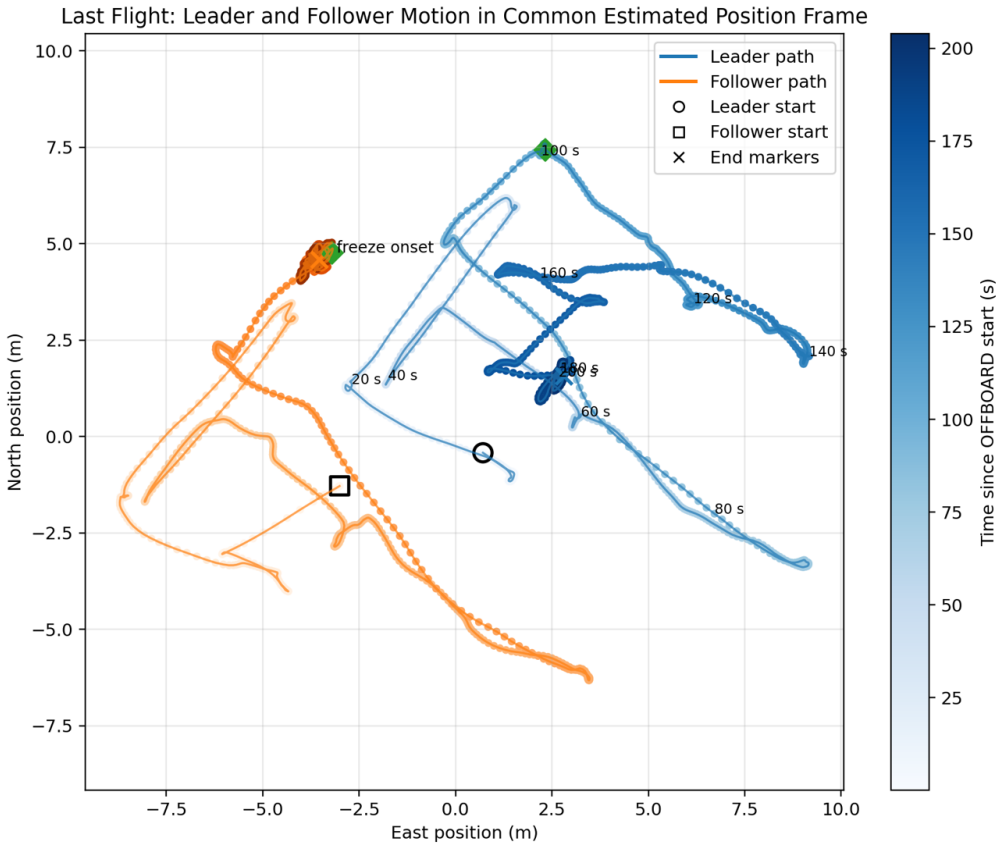


Figure 4.6: Representative trajectory from preliminary physical leader–follower testing. The follower generally tracked the commanded leader-relative motion, with delay attributed primarily to communication and control-loop latency.

Although stable leader–follower tracking was achieved during testing, the physical system remained sensitive to communication reliability. In several cases, OFFBOARD dropouts or failsafe triggers

occurred when the required setpoint stream or communication link was interrupted. This behavior is consistent with the requirements of PX4 OFFBOARD mode, which depends on continuous external setpoint updates. As a result, the physical testing highlighted communication continuity as one of the most important practical limitations of the current implementation. While the simulation environment allowed exact and uninterrupted state information, the real system depended on wireless telemetry, MAVLink routing, mavsdk server connections, and PX4 mode transitions all operating correctly at the same time.

The physical testing also provided an opportunity to verify portions of the haptic feedback pipeline. Full position-based formation-deformation and gate-avoidance haptic cueing was not completed in flight. However, bench testing was performed to verify that vehicle-state information could be routed through the software framework and converted into directional haptic commands. In particular, attitude-based roll cueing was tested to confirm that the haptic interface could receive state-dependent information and render the corresponding vibrotactile feedback to the operator. This supported the feasibility of integrating the same general haptic architecture used in simulation with the physical swarm-control framework.

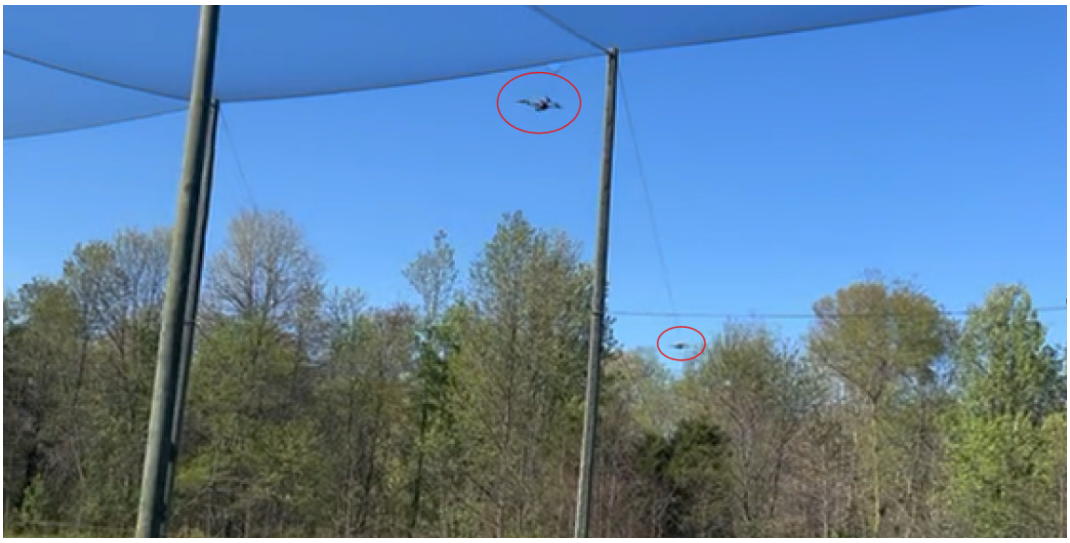


Figure 4.7: Representative video frame from preliminary physical testing at the Fearless Flight Facility.

Overall, the physical demonstration showed that the major elements of the proposed hardware implementation were functional. The custom quadcopters were able to fly under PX4 control, the ground-station computer was able to receive vehicle telemetry and maintain separate software interfaces for each drone, and the Python-based swarm controller was able to generate real-time follower commands. The follower was able to track the leader-relative setpoints during preliminary flight tests, demonstrating the basic feasibility of the centralized physical swarm architecture.

At the same time, these results should be interpreted as proof-of-implementation rather than a complete validation of the physical swarm experiment. The physical testing was limited to a reduced two-vehicle configuration, and full slalom-course testing with complete position-based haptic cueing was not completed. Communication reliability, OFFBOARD robustness, and position-estimate accuracy remained important limitations. Future physical testing should therefore focus on improving communication robustness, validating the full three-vehicle configuration, implementing the complete haptic cueing logic in flight, and conducting repeated trials over a physical slalom course.

Chapter 5: Conclusion

5.1 Overview of the Research

This thesis investigated whether haptic cues can improve pilot situational awareness and reduce overall mental workload during UAV swarm operation. In this work, a leader–follower swarm architecture was evaluated using a gated slalom course as the primary task scenario. The concept was first implemented in simulation, where a human-subject experiment was conducted with 25 participants. Following the simulation study, a similar architecture was implemented using custom-built and modified quadcopters. Although no formal physical experiment was conducted, the quadcopters were used to verify key portions of the system, including the communication architecture and haptic implementation. In both phases of the work, the broader goal was to help operators manage formation behavior, follower motion, and obstacle traversal without relying solely on visual information.

5.2 Summary of Main Findings

Analysis of the simulation results showed that haptic feedback generally improved formation-keeping performance. In trials with haptic cues enabled, pilots were better able to keep the follower vehicles closer to their nominal relative positions in both the lateral and longitudinal directions. In contrast, trials without haptic feedback generally showed larger deviations from the desired formation geometry. Gate-avoidance performance also showed some improvement when haptics were enabled, although the differences were noticeably more modest than those observed for formation keeping.

Visual condition appeared to have a much stronger effect on missed-gate performance than the presence or absence of haptic feedback. On average, performance in the degraded visual environment (DVE) was significantly worse than in the good visual environment (GVE), indicating that visibility

remained a major factor in successful gate traversal.

Haptic feedback did not appear to substantially increase workload, although some pilots noted that the system required a period of adjustment. At the same time, haptic cueing improved subjective situational awareness by giving pilots a clearer sense of follower position relative to the leader vehicle. In this way, the haptic system provided a useful supplementary source of swarm-state information that was not as easily available through vision alone.

5.3 Interpretation of Results

These results suggest that haptic feedback was especially useful for conveying information that was difficult for the pilot to perceive directly during the task. In particular, the cueing appeared to be most effective for communicating follower lead–lag behavior and overall formation distortion. Gate navigation, however, remained largely vision-dominated, especially in the degraded visual environment (DVE). This indicates that haptic feedback was most valuable as a supplementary information channel rather than as a replacement for visual guidance. This distinction is especially important in the present application, since the operator was supervising a swarm and monitoring follower behavior, rather than simply piloting a single vehicle.

5.4 Contribution of the Physical Demonstration

Although the physical implementation was not the primary form of evaluation, it provided an important practical extension of the simulation framework. The physical demonstration showed that the overall control concept could be translated to real hardware using custom-built quadcopters and a centralized leader–follower architecture. Within this framework, the communication scheme was

successfully implemented and tested, and portions of the haptic system were also evaluated during ground testing. As a result, the physical work served as a proof of feasibility and helped bridge the gap between the simulation study and future experimental swarm testing.

5.5 Limitations

Although the simulation portion provided a refined and well-controlled testing environment, there were still areas in which the overall approach could be improved. One such area was the tuning of the haptic gate-boundary cueing logic. Several subjects noted that the gate-related cues occasionally arrived too late to be fully useful, particularly during faster or more aggressive maneuvering. At the same time, earlier versions of the cueing logic sometimes produced feedback too early or in situations where corrective action was not yet necessary. This highlights the difficulty of selecting haptic activation bounds that are both timely and intuitive for a wide range of pilot behaviors.

The physical portion introduced the greatest limitations, and the strongest conclusions of this thesis therefore come from the simulation study rather than the hardware implementation. The physical testing was performed with a two-drone swarm, whereas the intended long-term configuration consists of three vehicles. In addition, the physical system relied on a centralized control architecture, which introduced communication-related limitations because telemetry had to be transmitted to the ground station before follower commands could be computed and sent back to the vehicles. This added latency and increased the possibility of communication dropouts or delayed responses during operation.

The most significant limitation of the physical setup, however, was the accuracy of the recorded vehicle positions. Because the formation controller depended on the estimated states of the vehicles, position error directly affected tracking performance and the quality of the resulting swarm behavior.

Inaccuracies in the measured positions therefore made it more difficult to distinguish true control limitations from sensing and state-estimation limitations.

More broadly, the slalom task provided a structured and useful test scenario, but it does not represent every type of swarm mission or operational environment. In addition, although the human-subject experiment provided meaningful trends, participant-to-participant variability and the limited sample size of the final analysis set should still be considered when interpreting the broader generality of the results.

5.6 Recommendations for Future Work

There are several natural directions for future work based on the results and limitations of this thesis. The most immediate next step is more extensive physical flight testing with repeated trials and a full three-drone swarm. This would allow the same general ideas explored in simulation to be evaluated more completely on the real system and would make it possible to compare physical swarm performance across multiple runs rather than only verifying implementation feasibility.

Another important direction is improvement of the localization and state-estimation methods used for the physical platform. Since tracking performance was strongly affected by position accuracy, future work would benefit from more precise methods such as RTK GPS, motion-capture systems, relative ranging, or onboard vision-based approaches. Improving the quality of the state estimates would make it easier to evaluate the actual performance of the swarm controller rather than the limitations of the sensing architecture.

Future work could also explore larger swarms and more complex maneuvering tasks. While the gated slalom provided a useful structured benchmark, it represents only one type of swarm task.

Additional scenarios involving tighter maneuvering, different obstacle layouts, or more dynamic trajectories would help determine how well the haptic cueing strategy generalizes to broader human–swarm interaction problems.

The haptic feedback strategy itself also offers several opportunities for refinement. Alternative cue designs could be investigated, including more predictive cueing laws, adaptive cue thresholds, or different mappings between swarm-state errors and haptic output. In addition, it may be valuable to study multimodal feedback strategies in which haptics are combined with other forms of operator support, such as audio or improved visual displays.

Finally, future work could compare different swarm-control architectures and operator populations. For example, centralized, decentralized, and hybrid control strategies may each place different demands on the operator and may interact with haptic feedback in different ways. Similarly, testing with operators of different training levels or prior drone experience could provide further insight into how broadly these findings apply.

5.7 Closing Remarks

Overall, the results of this thesis show that vibrotactile haptic cueing is a promising method for improving human supervision of a UAV swarm. The strongest benefits appeared in formation awareness and formation-keeping performance, where haptic feedback helped pilots better understand the behavior of the follower vehicles without adding substantial workload. The simulation study provided the primary evidence for these conclusions, while the physical implementation demonstrated that the same general concept can be translated to a real hardware platform. Taken together, these results support the continued development of haptic feedback as a useful tool for future human–swarm

interaction systems.

Appendix A: Subjective Rating Scales

This appendix provides the subjective rating scales used during the simulation experiment for mental workload and situational awareness assessment.

A.1 Bedford Workload Scale

The Bedford Workload Scale used during the simulation experiment is shown in Fig. A.1.

A.2 China Lake Situational Awareness Rating Scale

The China Lake Situational Awareness Rating Scale used during the simulation experiment is shown in Fig. A.2.

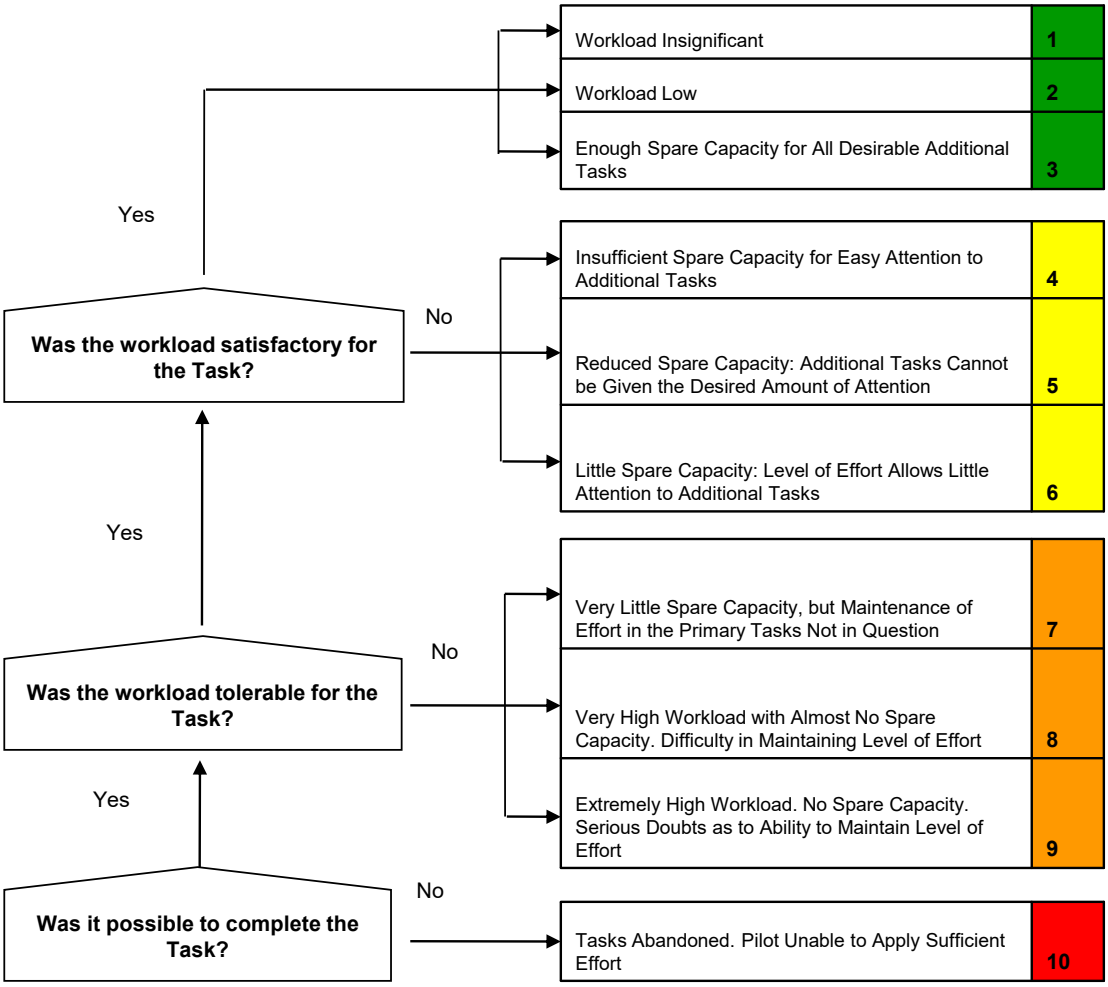


Figure A.1: Bedford Workload Scale used during the simulation experiment.

SA SCALE VALUE	CONTENT
VERY GOOD 1	• Full knowledge of a/c energy state/ tactical environment/ mission; • Full ability to anticipate/ accommodate trends
GOOD 2	• Full knowledge of a/c energy state/ tactical environment/ mission; • Partial ability to anticipate/ accommodate trends; • No task shedding
ADEQUATE 3	• Full knowledge of a/c energy state/ tactical environment/ mission; • Saturated ability to anticipate/ accommodate trends; • Some shedding of minor tasks.
POOR 4	• Fair knowledge of a/c energy state/ tactical environment/ mission; • Saturated ability to anticipate/ accommodate trends; • Shedding of all minor tasks as well as many not essential to flight safety/ mission effectiveness
VERY POOR 5	• Minimal knowledge of a/c energy state/ tactical environment/ mission; • Oversaturated ability to anticipate/ accommodate trends; • Shedding of all tasks not absolutely essential to flight safety/ mission effectiveness.

Figure A.2: China Lake Situational Awareness Rating Scale used during the simulation experiment.

Bibliography

[1] M. T. Morcos, S. M. Fishman, A. Cocco, U. Saetti, T. Berger, M. Godfroy-Cooper, and E. N. Bachelder. Full-body haptic cueing algorithms for augmented pilot perception in degraded/denied visual environments. In *Proceedings of the 79th Annual Forum of the Vertical Flight Society*, West Palm Beach, FL, May 15-18, 2023. doi: 10.4050/F-0079-2023-18072.

[2] RadioMaster RC. TX16S Mark II Radio Controller (Mode 2). https://radiomasterrc.com/products/tx16s-mark-ii-radio-controller, n.d. Accessed: 2026-04-28.

[3] Varjo. Varjo and OpenBCI Partner to Bring Neurotechnology to Spatial Computing. https://varjo.com/news/openbci-and-varjo-partner-to-bring-neurotechnology-to-spatial-computin May 2022. Accessed: 2026-04-28.

[4] M. T. Morcos, S. M. Fishman, U. Saetti, T., M. Godfroy-Cooper, and E. N. Bachelder. Full-body haptic and spatial audio cueing algorithms for augmented pilot perception. In *Proceedings of the 80th Annual Forum of the Vertical Flight Society*, Montreal, Quebec, Canada, May 7-9, 2024. doi: 10.4050/F-0080-2024-1179.

[5] Michael T. Morcos, Spencer Fishman, Umberto Saetti, Tom Berger, Edward N. Bachelder, and Martine Godfroy-Cooper. Full-body haptic and spatial audio cueing for augmented pilot perception. *Journal of Guidance, Control, and Dynamics*, 48(6):1334–1347, 2025. doi: 10.2514/1.G008722.

[6] Anon. Aeronautical design standard performance specification handling qualities requirements for military rotorcraft. Technical report, U.S. Army Aviation and Missile Command Aviation Engineering Directorate, Redstone Arsenal, AL, 2000.

[7] 3DR Inc. 3dr store: Product catalog. https://store.3dr.com/, n.d. Accessed: 2026-04-16.

[8] Oscar Liang. Frsky r-xsr receiver overview. https://oscarliang.com/frsky-r-xsr-receiver/, n.d. Accessed: 2026-04-16.

[9] Rotorama. Jumper t-lite v2 elrs. https://www.rotorama.com/product/jumper-t-lite-v2-elrs, n.d. Accessed: 2026-04-16.

[10] S. MahmoudZadeh, A. Yazdani, Y. Kalantari, B. Ciftler, F. Aidarus, and M. O. Al Kadri. Holistic review of uav-centric situational awareness: Applications, limitations, and algorithmic challenges. *Robotics*, 13(8):117, 2024. doi: 10.3390/robotics13080117. URL https://doi.org/10.3390/robotics13080117.

[11] Vahid Hamdipoor and Youdan Kim. On the robust network design for manned-unmanned teaming (mum-t). In *Proceedings of the American Control Conference (ACC)*, pages —, 2022. doi: 10.23919/ACC53348.2022.9867428.

[12] Seonghyeon Kim and Youdan Kim. Development of an mum-t integrated simulation platform. *IEEE Access*, 11:21519–21533, 2023. doi: 10.1109/ACCESS.2023.3248096.

[13] M. Campion, P. Ranganathan, and S. Faruque. Uav swarm communication and control architectures: A review. *Journal of Unmanned Vehicle Systems*, 7(2):93–106, 2019. doi: 10.1139/juvs-2018-0008.

[14] K. K. Oh, M. C. Park, and H. S. Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015. doi: 10.1016/j.automatica.2014.10.022.

[15] N. Wang, J. Dai, and J. Ying. Uav formation obstacle avoidance control algorithm based on improved artificial potential field and consensus. *International Journal of Aeronautical and Space Sciences*, 22:1413–1427, 2021. doi: 10.1007/s42405-021-00407-6.

[16] B. D. O. Anderson, C. Yu, B. Fidan, and J. M. Hendrickx. Control of directed formations with a leader-first follower structure. *Automatica*, 44(5):1191–1198, 2008. doi: 10.1016/j.automatica.2007.09.004.

[17] U. Saetti, J. Enciu, and J. F. Horn. Flight dynamics and control of an evtol concept aircraft with a propeller-driven rotor. *Journal of the American Helicopter Society*, 67:032012, 2022. doi: 10.4050/JAHS.67.032012.

[18] Wei Ren and Randal W. Beard. Distributed consensus in multi-vehicle cooperative control. *Springer*, 2007.

[19] Amy Hocraffer and Chang S. Nam. A meta-analysis of human-system interfaces in unmanned aerial vehicle (uav) swarm management. *Applied Ergonomics*, 58:66–80, 2017. ISSN 0003-6870. doi: https://doi.org/10.1016/j.apergo.2016.05.011. URL https://www.sciencedirect.com/science/article/pii/S0003687016300989.

[20] Dominic J. Duxbury, Norman W. Paton, and John A. Keane. Dynamic decision making for situational awareness using drones: Requirements, identification and comparison of decision support methods. *Expert Systems with Applications*, 252:124057, 2024. doi: 10.1016/j.eswa.2024.124057.

[21] Wenjuan Zhang, Yunmei Liu, and David B. Kaber. Effect of interface design on cognitive workload in unmanned aerial vehicle control. *International Journal of Human-Computer Studies*, 189:103287, 2024. ISSN 1071-5819. doi: https://doi.org/10.1016/j.ijhcs.2024.103287. URL https://www.sciencedirect.com/science/article/pii/S1071581924000715.

[22] M. T. Morcos, G. Luzzani, M. R. Fischer, D. Jun, E. Gary, U. Saetti, D. H. Geiger, S. T. Kubik, A. R. Breed, and C. J. Crane. Full-body haptic cueing for augmented pilot perception in position/velocity tracking tasks. In *Proceedings of the* 81^{st} *Annual Forum of the Vertical Flight Society*, Virginia Beach, VA, May 20–22, 2025. doi: https://doi.org/10.4050/F-0081-2025-136.

[23] Heather Culbertson, Samuel B. Schorr, and Allison M. Okamura. Haptics: The present and future of artificial touch sensation. *Annual Review of Control, Robotics, and Autonomous Systems*, 1: 385–409, 2018. doi: 10.1146/annurev-control-060117-105043.

[24] Lynette A. Jones and Nadine B. Sarter. Tactile displays: Guidance for their design and application. *Human Factors*, 50(1):90–111, 2009. doi: 10.1518/001872008X250638.

[25] Jan B. F. Van Erp. Presenting directions with a vibrotactile torso display. *Ergonomics*, 48(3): 302–313, 2005. doi: 10.1080/0014013042000325725.

[26] Mario Olivari, Frank M. Nieuwenhuizen, Heinrich H. Bülthoff, and Lorenzo Pollini. Pilot adaptation to different classes of haptic aids in tracking tasks. *Journal of Guidance, Control, and Dynamics*, 37(6):1741–1753, 2014. doi: 10.2514/1.G000534.

[27] Allison M. Okamura. Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot*, 31(6):499–508, 2004. doi: 10.1108/01439910410566362.

[28] R. Bergholz and colleagues. Haptic feedback in robot-assisted surgery: Current state and future directions. *Frontiers in Robotics and AI*, 10:1183165, 2023. doi: 10.3389/frobt.2023.1183165.

[29] D. Leonardis, M. Solazzi, and A. Frisoli. Combining tactile and kinesthetic feedback for teleoperation of robotic systems. *IEEE Transactions on Haptics*, 17(1):54–66, 2024. doi: 10.1109/TOH.2023.3302912.

[30] T. Berger, M. B. Tischler, and J. F. Horn. Outer-loop control design and simulation handling qualities assessment for a coaxial-compound helicopter and tiltrotor. In *Proceedings of the 77th Annual Forum of the Vertical Flight Society*, Oct 5-8, 2020.

[31] T. Berger, M. B. Tischler, and J. F. Horn. High-speed rotorcraft pitch axis response type investigation. In *Proceedings of the 77th Annual Forum of the Vertical Flight Society*, May 10-14, 2021.

[32] C. M. Ivler, S. A. Tobias, K. Cunneen, G. L. Sinsley, C. W. Rowley, J. Watkinson, and F. Gandhi. Toward a uas handling qualities specification: Development of uas-specific mtes. In *Proceedings of the Vertical Flight Society's 78th Annual Forum*, Fort Worth, Texas, USA, 2022. Vertical Flight Society.

[33] C. M. Ivler, E. S. Rowe, J. Martin, J. S. Lopez, and M. B. Tischler. System identification guidance for multirotor aircraft: Dynamic scaling and test techniques. *Journal of the American Helicopter Society*, 66:022006, 2021. doi: 10.4050/JAHS.66.022006.

[34] J. G. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge University Press, 2006.

[35] Lorenz Meier, Dominik Honegger, and Marc Pollefeys. Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[36] Anis Koubaa et al. Micro air vehicle link (mavlink) in a nutshell: A survey. *IEEE Access*, 7: 87658–87680, 2019.

[37] Martin Saska, Tomas Baca, et al. Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles. *Journal of Intelligent & Robotic Systems*, 84(1-4):469–492, 2016.

[38] Matthew Turpin, Nathan Michael, and Vijay Kumar. Trajectory design and assignment for aggressive formation flight with quadrotors. In *Robotics: Science and Systems (RSS)*, 2014.

[39] Nathan Michael, Daniel Mellinger, Quentin Lindsey, and Vijay Kumar. The grasp multiple micro-uav testbed. In *IEEE Robotics and Automation Magazine*, volume 17, pages 56–65, 2010.

[40] Federico Augugliaro, Antonis P. Schoellig, and Raffaello D'Andrea. Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[41] Knoxlabs. bHaptics TactSuit X40: Advanced Haptic Suit with 40 Immersive Feedback Points. `https://www.knoxlabs.com/products/tactsuit-x40-vr-haptic-suit`, n.d.. Accessed: 2026-04-28.

[42] Knoxlabs. bHaptics Tactosy for Arms. `https://www.knoxlabs.com/products/tactosy-for-arm`, n.d.. Accessed: 2026-04-28.