Design and Implementation of a User-Friendly Interface and Pre-Flight Checklist Module on MotoGrid GCS for UAV Operations

Mohammad Reihan Maulana Dzakya), Merinda Lestandyb), Nur Kasanc), Muh. Akhirul Ramadhand), Ahmad Nadhif Ramadhani e), Daniel Saputra K f)

Department of Electrical Engineering, University of Muhammadiyah Malang, Jl. Raya Tlogomas No.246, Malang, Indonesia

b⁾Corresponding author: merindalestandy@umm.ac.id

a)maulanarehan478@gmail.com

**Abstract.** Unmanned Aerial Vehicle (UAV) operations are often constrained by the complexity of existing Ground Control Station (GCS) software interfaces. Popular platforms such as QGroundControl, although comprehensive, have a steep learning curve and potentially increase the risk of human error. This study aims to design, develop, and evaluate a new GCS called MotoGrid GCS, focusing on usability and intuitive operation management. The research adopts a Software Engineering approach combined with Research and Development (R&D) methodology, adapting the ADDIE model (Analysis, Design, Development, Implementation, Evaluation). The development process includes user needs analysis, C++ and QML-based architecture design, and prototyping for rapid iteration. The resulting system integrates four main modules, including an easy-to-understand user interface and an automated pre-flight checklist module. Functional and usability testing with 20 participants demonstrated high satisfaction in terms of ease of use, stability, and functionality. Results indicate that MotoGrid GCS is a safer and more efficient solution, particularly for beginner UAV operators. The main contributions of this research include the development of a user-friendly interface specifically tailored for UAV ground control operations, the integration of a systematic pre-flight checklist module to enhance safety compliance, the implementation of a modular architecture enabling seamless integration with existing MotoGrid functionalities, and empirical evaluation proving improved operational efficiency and checklist adherence.

# INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have experienced significant technological growth over the past decade, with applications spanning aerial mapping, environmental monitoring, precision agriculture, search and rescue, and defense [1], [2]. These diverse applications require not only advanced flight capabilities but also reliable and user-friendly Ground Control Station (GCS) systems to ensure efficient mission planning, monitoring, and control[3]. A GCS serves as the primary interface between UAV operators and the aircraft, integrating telemetry, navigation, and mission management into a single operational environment[4].

However, despite advancements in GCS technology, many existing platforms such as QGroundControl and Mission Planner, while feature-rich, present a steep learning curve for novice users[5], [6]. The complexity of their interfaces and the abundance of non-essential controls can overwhelm operators, particularly beginners, thereby increasing the risk of operational errors[7]. Furthermore, the absence of integrated and systematic pre-flight safety procedures in many platforms often leads to overlooked checks, potentially compromising both mission safety and UAV integrity [8].

Previous research in this field has largely focused on enhancing UAV flight performance, sensor integration, and autonomous capabilities, while paying less attention to improving the usability and operational intuitiveness of GCS software [9]. For example, Ramírez-Atencia and Camacho [5] and Politowicz *et al.* [3] demonstrated advanced mission planning and multi-vehicle control capabilities, but these solutions often sacrificed simplicity in user interaction. While platforms such as QGroundControl and Mission Planner provide comprehensive functionalities, their complexity can become a barrier for new operators [7].

Studies on UAV interface usability, such as those by Tabassum *et al.* [8] and Nelson and Johnson [9], emphasize the importance of user-centered design in reducing cognitive load and improving mission safety. Recommendations from these studies include clear visual layouts, intuitive navigation, and the integration of safety checklists [4], [10]. Modular interface customization efforts, such as those seen in MAVProxy extensions [5], still lack the inclusion of automated pre-flight safety procedures in open-source GCS solutions. To address these limitations, this research introduces MotoGrid GCS, a next-generation ground control station that integrates a user-friendly interface with an automated pre-flight checklist module. The system is developed using a Research and Development (R&D) methodology guided by the ADDIE model and supported by rapid prototyping for iterative refinement[2].

# literature review

## Intuitive and Easy-to-Understand Interface Design

The development of intuitive and easy-to-understand user interfaces is closely tied to the usability principles defined in the ISO 9241-11:2018 standard, which emphasizes three core aspects: effectiveness, efficiency, and user satisfaction[1]. This standard, widely adopted in both industry and academia, provides measurable criteria for determining whether a system can be operated efficiently by its target users. In the context of UAV GCS, adherence to this standard means that interface design must support quick learning curves, minimize the possibility of operational errors, and provide an overall positive user experience.

In Indonesia, research conducted from 2021–2024 has demonstrated the successful application of the User-Centered Design (UCD) methodology in the development of e-learning platforms and business systems, achieving System Usability Scale (SUS) scores ranging from 75–83% [2], [3]. These results show that involving users throughout the design process directly impacts usability outcomes. By drawing from these findings, a UAV GCS designed with UCD principles can ensure that essential functions are presented clearly, with minimal clutter and optimized task flows, ultimately increasing operator confidence during real missions.

The emergence of the UX 3.0 paradigm and the Human-Centered AI approach has further influenced interface design strategies in the AI era[4], [5]. UX 3.0 shifts the design focus from isolated usability metrics toward the *experience ecosystem*, where system interactions are considered holistically across devices, environments, and user mental states. Human-Centered AI complements this by ensuring that intelligent systems, including AI-assisted GCS, maintain transparency, explainability, and trustworthiness. For UAV operations, this means that while AI may assist in decision-making, the operator always retains situational awareness and control.

## Pre-Flight Checklist

Pre-flight safety procedures are a critical component in ensuring mission success and UAV operational integrity. A pre-flight checklist serves to verify that all system components, sensors, communication links, and environmental conditions are within operational limits before takeoff[6]. In manned aviation, checklist adherence is a well-established practice, but in UAV operations—especially in civilian and commercial use—the implementation is often manual and prone to oversight.

Studies in UAV safety management have shown that automated pre-flight checklists significantly reduce human error rates by enforcing step-by-step validations and preventing mission initiation until all safety criteria are met [7], [8]. Modern GCS platforms have attempted to integrate checklist features, but most are either non-interactive or rely on operator discretion. Incorporating an automated and interactive pre-flight checklist module not only standardizes safety procedures but also improves operational readiness by minimizing cognitive load during mission preparation.

**Human Factors and Modular Architecture in UAV GCS Design**

Human factors engineering plays a crucial role in UAV GCS design, focusing on reducing cognitive workload, improving situational awareness, and enhancing decision-making efficiency [8]. Cognitive overload in UAV operations can lead to delayed responses, misinterpretation of data, or procedural errors. Therefore, GCS interfaces should balance the presentation of critical information with the suppression of unnecessary details.

A modular architecture approach in GCS development supports scalability, flexibility, and maintainability[9]. By designing independent but interoperable modules—such as mission planning, telemetry monitoring, and safety management—developers can update or replace functionalities without disrupting the entire system. This approach also enables the integration of emerging technologies, such as AI-based decision support and autonomous flight control, without requiring a complete redesign. Combining modular architecture with human factors engineering principles ensures that the GCS remains both technologically advanced and operator-friendly.

# RESEARCH METHOD

This study employs a Software Engineering approach combined with a Research and Development (R&D) model to ensure systematic and iterative development of the MotoGrid GCS system. The R&D approach is selected to facilitate the continuous refinement of the software through successive cycles of design, prototyping, testing, and evaluation, ensuring that each development stage directly addresses user needs and operational requirements. To guide the process, the study adopts the ADDIE model—an instructional design framework consisting of Analysis, Design, Development, Implementation, and Evaluation—due to its structured and measurable nature in managing complex software projects.

In the Analysis phase, the focus is on gathering and analyzing user requirements, operational constraints, and UAV mission needs, particularly those related to interface usability and pre-flight safety procedures. The Design phase translates these requirements into detailed system specifications, including architecture planning using C++ and QML, interface wireframing, and workflow mapping based on mission scenarios. The Development phase involves building the functional system through modular programming, integrating the user-friendly interface with the automated pre-flight checklist module, and applying a prototyping methodology to allow for rapid iteration and early feedback.

During the Implementation phase, the system is deployed in a controlled environment and tested with real UAV operations to validate its functionality, performance, and usability. Finally, the Evaluation phase assesses the system against predefined criteria, such as the System Usability Scale (SUS), operational error rate reduction, and overall user satisfaction, involving both novice and experienced UAV operators. This structured methodology ensures that the final product not only meets technical specifications but also aligns with user expectations, safety standards, and operational efficiency goals in UAV mission management.

## Analysis

This stage aimed to identify both the functional and non-functional requirements of the MotoGrid GCS system as a foundation for its development. The process began with a comprehensive literature study that examined the official documentation of QGroundControl, relevant drone flight regulations issued by international and national aviation authorities such as ICAO, EASA, FAA, and CASR (Indonesia), as well as established standards in usability and risk management. This step provided theoretical and regulatory insights to ensure that the system aligns with global best practices and legal compliance.

In parallel, field observations were conducted to study existing UAV operational workflows and geofencing mechanisms implemented in widely used platforms like DJI and ArduPilot. These observations allowed the research team to understand practical challenges, system limitations, and effective interface strategies used in real-world UAV operations.

Furthermore, interviews were held with UAV operators and field technicians to gather direct input regarding their operational experiences, pain points, and expectations for an improved GCS. The combination of literature review, on-site observation, and expert interviews provided a comprehensive understanding of user requirements, ensuring that the system design would be grounded in both regulatory standards and real operational needs.

## Design

The system was developed as a set of interconnected features integrated within MotoGrid GCS, designed to provide seamless interaction between the user interface, core functionalities, and safety mechanisms. The frontend was built using QML (Qt Quick) to deliver an intuitive and easy-to-understand user interface, ensuring that operators can navigate and execute UAV operations efficiently, even with minimal prior experience. This design approach emphasizes clarity, responsiveness, and accessibility, aligning with modern usability standards.

The backend was implemented using C++ to handle the system’s business logic, information management, and the processing of no-fly zone data in GeoJSON format. This architecture enables the GCS to efficiently process spatial data, validate flight routes, and enforce geofencing restrictions in real time, thereby enhancing operational safety. The backend and frontend communicate seamlessly, ensuring that any updates to mission parameters or safety constraints are immediately reflected in the interface.

To ensure a coherent and well-documented development process, visual design tools were employed throughout the project. UML diagrams—including Use Case, Activity, and Class diagrams—were utilized to model the system’s workflow, interactions, and structural relationships between components. Additionally, user interface mockups were created in Figma to provide a clear visual concept during the early design phase, facilitating feedback and iteration before implementation. This combination of structured backend logic, user-friendly interface design, and well-documented visual modeling ensured that MotoGrid GCS met both functional requirements and user experience goals.

## Development

To accelerate the development cycle and ensure that the final product aligned with user needs, the Prototyping method was adopted. This approach allowed for rapid iteration through a sequence of short development phases, beginning with quick planning to outline essential features, followed by the creation of a functional prototype. The prototype was then evaluated directly by UAV operators and field technicians, whose feedback was used to refine the system in successive cycles. Iteration continued until the prototype satisfied all predefined functional and non-functional requirements. The development process was carried out by cloning and modifying the open-source QGroundControl codebase, enabling the reuse of proven core functionalities while implementing significant modifications in the interface, pre-flight checklist module, and safety validation features to meet the specific objectives of MotoGrid GCS.

A diagram of a route

Description automatically generated

**FIGURE 1.** System Workflow Flowchart for MotoGrid GCS.

To provide a clearer understanding of the operational workflow, Figure 1 presents the System Workflow Flowchart for MotoGrid GCS. The diagram outlines the step-by-step sequence from system startup to the completion of data documentation. The workflow begins with the Start node, representing the initialization of the operational process. This is followed by the Start & Connect MotoGrid GCS step, where the software is launched and establishes a communication link with the UAV. Once the connection is confirmed, the system initiates the Pre-Flight Checklist to verify readiness, including checks on sensors, hardware, connectivity, and battery status.

Next, the operator proceeds to Create a Waypoint Route, defining the UAV’s flight path according to the mission objectives. This is followed by a Route in Restriction Area? decision node, where the system automatically validates the route against stored geofencing data. If the route intersects with a restricted area, the process loops back to the route creation step for revision (*Yes* branch). If the route passes validation (*No* branch), the process continues to Saving Flight Routes, where the approved flight path is stored in the system’s mission records. Finally, the workflow concludes with Data Storage and Documentation, ensuring that all mission details are archived for post-flight analysis, reporting, and compliance audits. The End node marks the completion of the workflow, signifying readiness for the UAV to execute the planned mission.

This structured flow not only standardizes UAV mission preparation but also reduces the risk of human error, ensures compliance with operational regulations, and maintains a complete operational record for accountability and future reference.

## Implementation

The system was developed and deployed within a specific technical environment to ensure optimal performance, compatibility, and ease of maintenance. The implementation utilized C++ and QML as the primary programming languages, combining robust backend processing capabilities with a flexible and interactive frontend design. Development was supported by a range of tools and frameworks, including Qt Creator as the integrated development environment, Qt 5.15 for cross-platform GUI development, CMake for build automation, and Git for version control and collaborative development.

The project was carried out across multiple operating systems to ensure cross-platform compatibility, with Windows 10 and Ubuntu 22.04 serving as the primary development environments. Prior to deployment in real UAV operations, the system underwent rigorous validation through a PX4 SITL (Software In The Loop) simulation environment. This simulation phase allowed the development team to test and verify functionality in a controlled and repeatable setting, reducing the risks associated with early-stage hardware testing and ensuring system stability before real-world trials.

## Evaluation

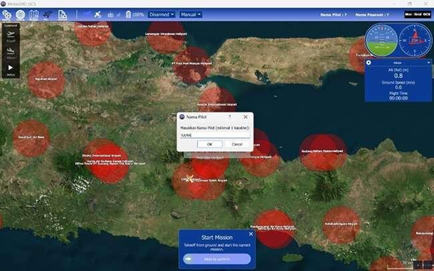
The evaluation phase was conducted using a combination of functional and performance testing to ensure that the MotoGrid GCS met both technical specifications and user expectations. Functional testing focused on verifying the correct operation of all implemented features, including the automated pre-flight checklist, No-Fly Zone warning system, and information logging module. Each function was tested against predefined acceptance criteria to confirm that it performed as intended without generating errors or inconsistencies during operation.

In addition to functional verification, performance testing was carried out to assess the system’s responsiveness and processing capabilities. Quantitative measurements were taken for parameters such as system response time and data processing speed under varying operational conditions. Complementing these technical metrics, qualitative evaluation was performed by collecting user feedback from UAV operators who interacted with the system. Their assessments provided valuable insights into the ease of use, intuitiveness of the interface, and perceived effectiveness of the implemented features, enabling further refinement to enhance overall user experience.

# RESULTS AND DISCUSSION

## Module Development Results

The research indicates that the UAV operations management module designed with MotoGrid GCS was successfully developed according to the system requirements. Key developed features include an easy-to-understand UI and an automated Pre-Flight Checklist.

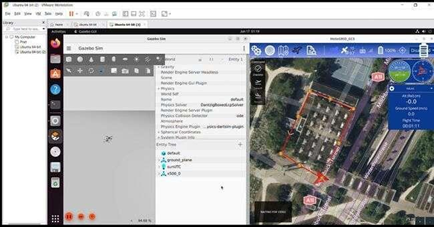


**FIGURE 2.** Initial Interface Of Motogrid Gcs

Figure 2 presents the initial interface of MotoGrid GCS, showcasing the integration of key operational features designed to enhance UAV mission management. The interface adopts a satellite map view as the primary navigation and planning environment, overlaid with clearly marked No-Fly Zones represented by red circular areas. This geospatial visualization enables operators to easily identify restricted regions during mission planning, thereby reducing the risk of airspace violations.

At the center of the interface, a dialog box is displayed for entering the pilot’s name, which serves as part of the mission initialization process and contributes to proper logging and documentation. The lower section contains a Start Mission control panel, which prompts the user to confirm readiness before commencing UAV operations. This ensures that all preparatory steps, including route validation and pre-flight checks, are completed prior to execution.

On the upper section of the screen, navigation menus and operational controls are provided, including options for zooming, mission mode selection, and system status monitoring. On the right side, a live telemetry panel displays real-time flight data such as altitude, ground speed, and GPS coordinates, allowing the operator to maintain situational awareness throughout the mission. The combination of intuitive visual elements, clear geographic overlays, and integrated safety prompts reflects the system’s emphasis on usability, operational safety, and compliance with aviation regulations.

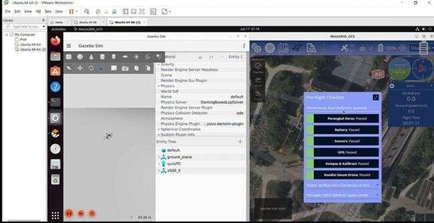


**FIGURE 3.** Drone Simulation Using Ubuntu.

Figure 3 illustrates the UAV simulation environment used during the testing phase of MotoGrid GCS, conducted on Ubuntu as the primary operating system for system validation. On the left side of the screen, the Gazebo Simulator interface is displayed, showing the simulation workspace along with the available UAV models and associated plugin configurations. Gazebo was utilized to emulate realistic flight dynamics and environmental interactions, enabling safe and repeatable testing of the system without the risks associated with early hardware trials.

On the right side, the MotoGrid GCS interface is shown running concurrently, presenting a satellite map view of the simulated flight area. The map includes an outlined polygon marking the designated flight zone, which is critical for testing geofencing features and ensuring that the UAV remains within the allowed operational boundaries. Real-time telemetry data, including altitude, ground speed, and GPS coordinates, is displayed on the right panel of the GCS, allowing operators to monitor simulated flight parameters as they would during actual missions.

This integrated simulation setup demonstrates the interoperability between MotoGrid GCS and PX4 SITL through Gazebo, allowing the research team to validate both functional features—such as waypoint navigation and No-Fly Zone detection—and performance aspects like system responsiveness and data accuracy. By conducting these trials in a controlled simulation environment, the development team was able to refine the software before deploying it to real-world UAV operations, thereby reducing potential risks and improving overall reliability.



**FIGURE 4.** Pre-Flight Checklist Feature for UAV or Drone

Figure 4 showcases the Pre-Flight Checklist feature integrated into MotoGrid GCS, demonstrated within the PX4 SITL simulation environment running on Ubuntu. On the left side, the Gazebo Simulator is active, representing the virtual UAV model and environment setup. On the right side, the MotoGrid GCS interface is displayed, presenting the Pre-Flight Checklist panel as part of the mission preparation workflow.

The checklist contains several critical system readiness checks, including Power Supply Status, Battery Status, Network Connectivity, GPS Lock, Compass and Sensor Calibration, and No-Fly Zone Verification. Each checklist item is accompanied by a visual status indicator that changes based on system validation results, allowing operators to quickly assess whether all required conditions for safe flight have been met. This structured verification process significantly reduces the likelihood of operational errors by ensuring that no critical step is overlooked prior to mission execution.

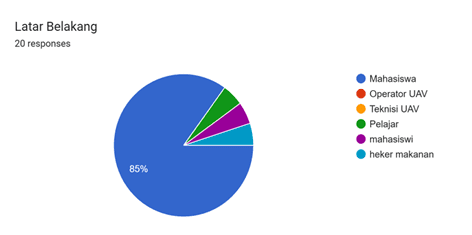
By integrating the Pre-Flight Checklist directly into the MotoGrid GCS interface, the system promotes a standardized and user-friendly approach to UAV operations. This feature not only enhances safety and compliance with aviation guidelines but also streamlines mission preparation, making it especially valuable for beginner operators who may not yet be familiar with all pre-flight requirements.

The following table summarizes the main components of the module and their functionality:

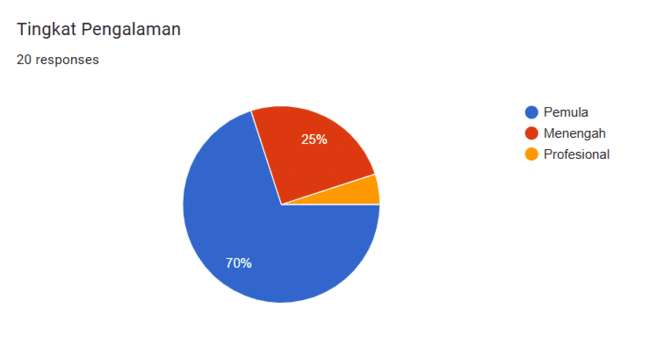
**TABLE 1.** Features and Functions of the MotoGrid GCS Application.

|  |  |  |
| --- | --- | --- |
| **No** | **Feature Module** | **Main Function** |
| 1 | Design UI | An easy-to-understand GUI for beginners |
| 2 | Pre-Flight Checklist | Displays checks for each part of the aircraft |

Of the 20 people who tested MotoGrid GCS, 85% were from a student background, and the number of new or novice users was 70%. These results can be seen in the graphs below:



**FIGURE 5.** Background of the Users.



**FIGURE 6.** User Experience Level with GCS.

Then, from the user experience during the application's use, a graph was obtained showing satisfaction in a single mission flight trial:

|  |  |
| --- | --- |
| Forms response chart. Question title:  Antarmuka aplikasi MotoGrid GCS mudah dipahami . Number of responses: 20 responses.  (a)  Forms response chart. Question title: Aplikasi stabil dan jarang mengalami crash atau error  . Number of responses: 20 responses.  (c) | Forms response chart. Question title: Waktu respon aplikasi terhadap perintah pengguna cukup cepat  . Number of responses: 20 responses.  (b)  Forms response chart. Question title:   Fitur-fitur yang ada berjalan sesuai ekspektasi  . Number of responses: 20 responses.  (d) |

**FIGURE 7.** Satisfaction Graph for the MotoGrid GCS Application System.

## Discussion

This research successfully created a flexible module compatible with the MotoGrid GCS platform to effectively manage UAV operations. The results show that this system can perform planning, monitoring, and reporting functions directly. This success is demonstrated by the modular strategy used in software creation and the utilization of the proven MAVLink library, which can function with many autopilot systems. Furthermore, the integration of open-source maps (such as Leaflet and Mapbox) gives the system the ability to display maps and restricted areas with a high degree of accuracy. The results of this study reinforce the statement made by Widodo et al. (2021) who argued that the flexibility level of a GCS is determined by how open its communication protocol is and its modular structure. However, this study adds to that knowledge by including the aspect of risk management through the implementation of automatic marking on no-fly zones, which has not been discussed in detail in previous research. Theoretically, this advancement provides more value to the GCS management framework by incorporating elements of airspace control, making it more suitable for UAV use in complex locations such as urban areas. This module can be used as a reference for the development of other open-source GCSs

## Theoretical and Implementation Implications

Theoretically, this module emphasizes that a GCS system should act not only as a flight controller but also as a device for overall mission management. The practical consequences of this research are:

* Assisting UAV operators in managing missions without requiring other systems.
* Applicable in various missions, such as surveillance, mapping, and logistics in restricted areas.

# CONCLUSION

The MotoGrid GCS system was successfully developed with a flight logging feature equipped with metadata such as pilot identity and UAV ID, enabling more structured and easily traceable data documentation. The data logging feature in Sqlite (.db) format facilitates further analysis, reporting, and integration with other documentation systems. Functional testing showed that every system component runs according to specifications, including automatic data storage and data retrieval through the GUI. In terms of advantages, this system offers user-friendliness for novice operators, better data transparency, and flight history tracking capabilities. However, the system still has limitations, such as the lack of an integrated automatic cloud backup feature and minimal protection against operator metadata input errors.

# References

[1] D. E. Widiyanti, K. Asmoro, and S. Y. Shin, “HoloGCS: mixed reality-based ground control station for unmanned aerial vehicle,” *Virtual Real*, vol. 28, no. 1, Mar. 2024, doi: 10.1007/s10055-023-00914-9.

[2] J. Paterson *et al.*, “Improving Usability, Efficiency, and Safety of UAV Path Planning through a Virtual Reality Interface,” Apr. 2019, [Online]. Available: http://arxiv.org/abs/1904.08593

[3] M. S. Politowicz, E. T. Chancey, B. K. Buck, J. R. Unverricht, and B. J. Petty, “MPATH (Measuring Performance for Autonomy Teaming with Humans) Ground Control Station: Design Approach and Initial Usability Results.”

[4] O. M. HUBSKY, “Analysis of User Interfaces for Ground Control Stations of Unmanned Aerial Vehicles,” *Kibernetika i vyčislitelʹnaâ tehnika*, vol. 217, no. 3(217), pp. 5–23, Sep. 2024, doi: 10.15407/kvt217.03.005.

[5] C. Ramirez-Atencia and D. Camacho, “Extending QGroundControl for automated mission planning of Uavs,” *Sensors (Switzerland)*, vol. 18, no. 7, Jul. 2018, doi: 10.3390/s18072339.

[6] S. R. Flaherty, L. Fern, T. Turpin, and S. Scheff, “Universal Ground Control Station (UGCS) joystick evaluation,” in *IEEE Aerospace Conference Proceedings*, 2012. doi: 10.1109/AERO.2012.6187347.

[7] T. Amida, J. Chung, S. Roy, and E. Simon, “EVALUATING UAV GROUND CONTROL STATION DESIGN USING AVAILABLE HUMAN FACTORS GUIDELINES AND STANDARDS.” [Online]. Available: http://www.ryerson.ca/~j3chung/

[8] A. Tabassum, H. Bai, and N. Fala, “A Study on Workload Assessment and Usability of Wind-Aware User Interface for Small Unmanned Aircraft System Remote Operations,” Sep. 2023, [Online]. Available: http://arxiv.org/abs/2309.04543

[9] S. Li, Q. Wu, B. Du, Y. Wang, and M. Chen, “Autonomous Maneuver Decision-Making of UCAV with Incomplete Information in Human-Computer Gaming,” *Drones*, vol. 7, no. 3, Mar. 2023, doi: 10.3390/drones7030157.

[10] S. Lee, Y. Kwon, J. Heo, and J. Kim, “GCS Interface Mission Load Reduction Design through Pilot Gaze Point Analysis,” *International Journal of Mechanical Engineering and Robotics Research*, vol. 11, no. 9, pp. 710–717, Sep. 2022, doi: 10.18178/ijmerr.11.9.710-717.