MPATH (Measuring Performance for Autonomy Teaming with Humans) Ground Control Station: Design Approach and Initial Usability Results

Michael S. Politowicz[1], Eric T. Chancey[2], and Bill K. Buck[3]
NASA Langley Research Center, Hampton, VA 23681, USA

James R. Unverricht[4]
National Institute of Aerospace, Hampton, VA 23666, USA

Bryan J. Petty[5]
NASA Langley Research Center, Hampton, VA 23681, USA

Envisioned future Advanced Air Mobility (AAM) operations will require a transition of aircraft command and control from onboard pilots to remote operators. The National Aeronautics and Space Administration (NASA) has developed a research ground control station (GCS) software called MPATH (Measuring Performance for Autonomy Teaming with Humans) to study the human factors of remote operators in a representative AAM environment, where small uncrewed aerial systems (sUAS; simulated or real) act as surrogates for larger AAM aircraft. A primary focus of the research being conducted with the MPATH GCS is scalability (i.e., one human managing multiple vehicles). MPATH has demonstrated to be a useful capability for human factors research and remote operations. Two initial usability studies and a multi-vehicle control assessment were recently conducted, and usability data and operator feedback were collected. Generally, participants rated MPATH high on usability and interface quality, whereas information quality was rated slightly lower. These results were supported by specific feedback. Several generalized GCS design recommendations are proposed based on the results and feedback from these flight activities. Future updates to MPATH will incorporate the proposed recommendations. In practice, these recommendations could be used by any GCS software designer or developer to promote usability and safety.

I. Introduction

The envisioned future of Advanced Air Mobility (AAM) will require a transition of aircraft command and control from onboard pilots to remote operators. The AAM ecosystem is expected to include a diverse set of vehicles and operations, such as small uncrewed aircraft system (sUAS; i.e., drone) operations [1], Urban Air Mobility (UAM; e.g., air-taxis) [2], and Regional Air Mobility (RAM) [3]. The markets for these subsets of AAM are currently in the early stages of development, but to reduce costs, meet rising demands, and reach mature operations, it is anticipated that remote operators capable of managing multiple vehicles simultaneously will be necessary.

---

[1] Research Aerospace Engineer, Crew Systems and Aviation Operations Branch, M/S 152, AIAA Member.
[2] Human Factors Psychologist, Crew Systems and Aviation Operations Branch, M/S 152.
[3] Research Aerospace Engineer, Crew Systems and Aviation Operations Branch, M/S 152, AIAA Senior Member.
[4] Human Factors Psychologist, Crew Systems and Aviation Operations Branch, M/S 152.
[5] Aerospace Engineer, Aeronautics Systems Engineering Branch, M/S 238.

Relocating from the flight deck to a remote facility will inherently alter the types of information available to an aircraft operator. For instance, pilots on board an aircraft have the advantage of out-the-window views and vestibular cues (i.e., motion) to supplement and substantiate vehicle sensor data, whereas operators in a remote facility are limited to only the vehicle sensor data and camera feeds displayed in the remote ground control station environment, which are prone to delays and dropouts. This change in available information has the potential to alter the situation awareness of remote operators in comparison to onboard pilots, which could impact performance and safety. Previous research has explored the potential human factors issues associated with remote aircraft operations, but the primary focus of this area of research has been military UAS operations [4], which do not necessarily reflect the operating environment and operator roles envisioned for AAM (see Ref. [5] for definitions and differences in pilot and operator roles). New vehicle types, flight profiles, and mission objectives have led to new technologies for remote operations that warrant additional analyses.

As the primary source of information for a remote operator, the ground control station (GCS) user interface for AAM vehicles is critical for ensuring safe and efficient flights. Currently, sUAS operations represent the most mature subset of AAM, and from the perspective of remote vehicle operations, these sUAS vehicles and associated GCSs can serve as useful surrogates for future UAM and RAM technologies. In this paper, we discuss a research GCS that has been developed by the National Aeronautics and Space Administration (NASA) to explore the human factors of remote operations using sUAS vehicles in a representative UAM environment.

II. MPATH Ground Control Station

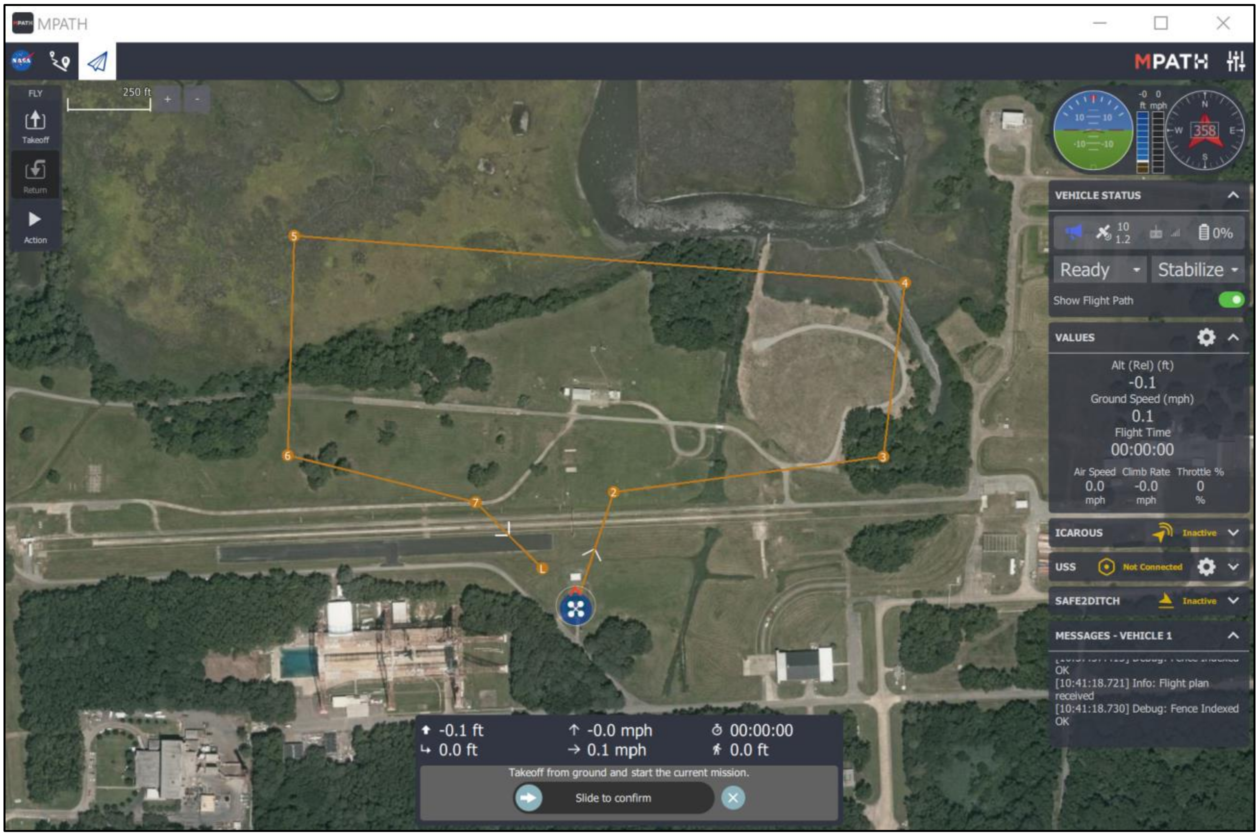
A. Background

With a broad focus on research, NASA is uniquely positioned to test novel aviation concepts and technologies. This research environment, which allows for extended visual line of site (EVLOS) sUAS operations, provides a level of safety that enables researchers to test operational limits without significant risk to people or property. These EVLOS operations permit the ground control station operator (GCSO) to be remotely located while trained observers in the field maintain visual contact with the vehicle to ensure safety of flight (see Ref. [6] for an example EVLOS configuration). Thus, researchers can evaluate safety-critical factors during routine remote operations rather than having to rely solely on post-hoc review of accident and incident reports.

Researchers at NASA Langley Research Center began exploring the human factors of remote sUAS GCSOs in a study that evaluated the effects of different autonomous traffic separation methods in a representative future AAM environment [7]. This 2019 study used an open-source GCS (without modifications) and a separate prototype traffic display interface. Participants in this study were tasked with managing only a single aircraft. Feedback from participants revealed several shortcomings of the GCS and a need to improve the display and integration of information presented to the GCSO.

The challenges of information processing for GCSOs in this environment will likely be amplified in future operations where a human operator may be tasked with managing more than one aircraft simultaneously from a single GCS (see Ref. [8]). In the face of these challenges, NASA's Transformational Tools and Technologies – Revolutionary Aviation Mobility (T^3-RAM) subproject has identified scalability of remote aircraft operations as a critical barrier for enabling mature AAM operations. Specifically, advances in research and technology will be required to allow few humans to operate many vehicles (referred to as *m:N*, in which *m* operators manage *N* vehicles [9]). The role of the human is expected to evolve as regulations and increasingly autonomous systems continue to mature [10], and it is important to understand the implications of these changes on human operators. Thus, to address the issue of scalability, T^3-RAM has identified human-autonomy teaming (HAT) as a critical area of research to be explored under the HAT Foundational Research Activity [11, 12].

Within the HAT Foundational Research Activity, NASA has developed the MPATH (Measuring Performance for Autonomy Teaming with Humans) GCS software to support remote sUAS operations and to conduct human factors research that enables multi-vehicle operations. The MPATH GCS is a software that is capable of controlling MAVLink-enabled sUAS (i.e., vehicles that use the MAVLink communication protocol [13]). MPATH is built upon the foundation of a widely used open-source GCS software (QGroundControl [14]), so it leverages existing functionality, safety, and testing while also providing a testbed for exploration of experimental features, such as HAT concepts and integration with advanced onboard automation. The MPATH GCS user interface is shown in Fig. 1. As the role of the remote aircraft operator evolves within the research environment, MPATH will also evolve to support the new roles, thus enabling test and validation of novel operational concepts in both simulation and flight. Furthermore, an experimental version of the MPATH GCS, called MPATHx (where *x* represents *Experimental*), will include exploratory features that can be exercised in simulation to assess forward-looking operational concepts and configurations that reach beyond the current operating limits of real aircraft.



Alt (Rel) (ft) -0.1
Ground Speed (mph) 0.1
Flight Time 00:00:00
Air Speed 0.0 mph
Climb Rate -0.0 mph
Throttle % 0 %

```
OK
[10:41:18.721] Info: Flight plan received
[10:41:18.730] Debug: Fence Indexed OK
```

-0.1 ft
0.0 ft
-0.0 mph
0.1 mph
00:00:00
0.0 ft

Takeoff from ground and start the current mission.

Fig. 1 MPATH Ground Control Station (v1.0.2) user interface.

B. Design Approach

For development of the MPATH GCS, we considered the following two approaches: developing the software either as an entirely new software application or as a modified version of an existing open-source software. In weighing these options, we established several requirements based on the intended operational environment and project constraints. The GCS software needed to be compatible with existing research sUAS and onboard research technologies, relatively low effort for development and maintenance, familiar to the GCSOs, and sufficiently reliable (i.e., stable and safe). Within the NASA research environment, the autopilot and research systems on board the vehicles use the MAVLink protocol [13], which is an open-source messaging protocol for communicating with sUAS. Because this protocol is widely used, several open-source MAVLink-based GCSs are available. The advantage of building upon one of these established open-source GCSs is the ability to leverage existing functionality, stability, testing, and updates from the open-source community. Additionally, there is potential for contributing fixes and features back to the open-source codebase. From the available options, we selected QGroundControl [14] as the foundation for the MPATH GCS. QGroundControl provides several unique and appealing features, such as the ability to install and run on multiple operating systems and a user interface that was designed for both mouse-based and touch-based interactions. Furthermore, the GCSOs at NASA Langley Research Center were already using QGroundControl to operate research vehicles, which promoted both familiarity and consistency when transitioning from one GCS to the other.

For the design and development of MPATH, we used a crowdsourcing method. Crowdsourcing provided access to a large community of design and development experts and the ability to engage unique portions of the community at each phase of the software development cycle. This method was not only efficient, but it included the added benefit of increased innovation, particularly in the user experience (UX) design phase. Both the design and development phases employed a challenge-based approach. During each challenge in the design phase, we were able to filter design submissions to our top three choices and then select features from each to be merged into the final design solution. This method allowed us to consider and integrate many creative solutions from designers with a variety of backgrounds. Across the design and development phases, there was a total of 45 challenges and 96 unique challenge participants.

C. Software Features

The studies described in this paper used MPATH version 1.0.2 (v1.0.2). This version represents an early stage of development for MPATH and includes only a limited set of new features, which are described in this section. Additional features have since been added to MPATH (e.g., improved multi-vehicle user interface), but those features are not discussed in this paper. Furthermore, several features included in MPATH v1.0.2 were not used in these initial studies, so those features are also excluded from this paper. MPATH v1.0.2 was built from QGroundControl version 4.1.3 (v4.1.3). The novel features included in MPATH v1.0.2 that extend beyond the functionality of QGroundControl are grouped into the following three categories: general user interface (UI) design improvements, improved automation transparency, and integrated human factors data collection capabilities.

Though many of the modifications described in this section are meant to address human factors concerns, the design of MPATH v1.0.2 is only an initial "best guess" established through our design process (described in the previous section). However, from this starting point, we have begun collecting empirical data directly from users that will support iterative and incremental improvements to the design. The first collection of empirical usability data and recommendations derived from these data are presented in the subsequent sections of this paper.

1. General UI Improvements

The user interface for the Fly View (i.e., the primary view used during flight; depicted in Fig. 1) was modified in several substantial ways to begin addressing general human factors concerns related to information access, to prepare for the integration of additional onboard and offboard systems (e.g., detect and avoid automation, flight scheduling servers) that will introduce more information into the display, and to prepare for multi-vehicle control from a single GCS. QGroundControl v4.1.3 generally consolidated flight relevant information in the top toolbar and in the Telemetry Values Bar at the bottom of the display. For MPATH, we established a modular widget-based design on the right side of the display and moved the information from the top toolbar to a new Vehicle Status widget on the right. The widgets in MPATH are collapsible, removable, and rearrangeable. Furthermore, new widgets can be developed as new data connections are implemented within MPATH. By consolidating flight relevant information, we are able to increase the spatial proximity of related information, which reduces scanning effort and facilitates more efficient information processing (see proximity compatibility principle [15]). Additionally, providing operators with the ability to customize the organization of widgets can help to maintain these advantages across mission and vehicle types, in which relevance and priority of information may vary.

Two important considerations should be mentioned regarding the implementation of these features in MPATH v1.0.2. First, we did not remove the Telemetry Values Bar at the bottom of the display (see Fig. 1) despite making this information available in the Values widget on the right, which directly opposes our justification for the widget-based design. There are two reasons for this design decision: the slider bar for confirming actions (i.e., a UI element that must remain prominent) is integrated into this bottom Telemetry Values Bar, and additionally, we wanted to gather feedback and eye movement data to determine user location preferences for these critical vehicle state data. Second, although the widgets were customizable (i.e., rearrangeable, removable) in this version of MPATH, the custom configuration would reset when the software application was closed, and configurations could not be saved by the user. Thus, users were required to rearrange the widgets before every flight, which inhibited the usability of this feature.

2. Automation Transparency

One of the key motivations for developing the MPATH GCS was integration with onboard automation. The following two specific automated systems were integrated with MPATH: ICAROUS (Independent Configurable Architecture for Reliable Operations of Unmanned Systems) and Safe2Ditch. ICAROUS is an open-source software developed by NASA that provides functionality for autonomously rerouting around other aircraft as well as geofenced areas [16]. Safe2Ditch is a crash management software that provides functionality to autonomously select a ditch site (from a predefined list of ditch sites), reroute to that site, and execute a landing [17]. Both of these automated systems were used in a previous study with an open-source GCS (not modified), and feedback from participants indicated a need for more insight into the behavior and intent of these systems [7].

Two widgets were developed within the MPATH GCS with the goal of improving automation transparency (i.e., availability of information about the automation's behavior and intent [18]) for these systems. The ICAROUS widget provides the overall system state (i.e., active or not active), conflict detection state (i.e., conflict with another vehicle detected or not detected), and status messages from the system. In addition to the widget, MPATH also displays heading bands around the vehicle icon, which are wedge shaped areas positioned within a circle around the vehicle that are color coded to indicate safe and unsafe directions of travel. MPATH also displays the new flight path created by ICAROUS when rerouting. The Safe2Ditch widget provides the overall system state (i.e., active or not active), auto-landing state (i.e., auto-landing engaged or not engaged), and a dynamic list of the top three ditch sites. In addition to the widget, MPATH also displays the ditch sites on the map and provides an interface for uploading ditch sites to Safe2Ditch.

An important consideration regarding these automation transparency features is that they require data flow from the automated systems on board the vehicle. However, this data flow is not possible on some vehicle configurations even when the automation is present and active on the vehicle, which was the case for the initial usability studies described in this paper. Without the correct vehicle configuration, the automation transparency features may be substantially limited, yet basic information about the automation (e.g., activation status) is typically still available.

3. *Human Factors Data Collection Capabilities*

To facilitate human factors data collection, a feature was added to the MPATH GCS to log user interactions as well as the dynamic screen positions of areas of interest (AOIs; i.e., regions of the display) and objects of interest (i.e., individual UI elements, such as a vehicle icon on the map). These screen positions are logged with timestamps to enable post-hoc association with eye gaze positions, which are collected using separate eye tracking software. This logging feature runs in the background and has no impact on the user's experience.

III. Initial Usability Studies

A. Method

In 2021 and 2022, the MPATH GCS was used to remotely operate aircraft during two flight activities: one with simulated vehicles and one with real vehicles (EVLOS operations). The simulated flight activity was completed over the span of one month and included six participants sampled from the population of GCSOs at NASA Langley Research Center. Each participant completed 17 simulated flight operations. For a detailed description of this simulated flight activity, see Ref. [19].

The EVLOS flight activity was completed over the span of five months and included seven participants who were also sampled from the population of GCSOs at NASA Langley Research Center. Both flight activities were conducted in the ROAM (Remote Operations for Autonomous Missions) UAS Operations Center in the Air Traffic Operations Lab (ATOL) at NASA Langley Research Center [20] (see Fig. 2). For a detailed description of this EVLOS flight activity, see Ref. [21].

Fig. 2 ROAM UAS Operations Center during EVLOS flight activity.

In both flight activities, participants were tasked with regular duties of a GCSO within the GCS application. These included loading the flight plan, launching the vehicle, observing automated flight, observing autonomous reroutes, and landing the vehicle. Both flight activities included scenarios where the GCSO had to interact with specific automated systems on board the vehicle. For ICAROUS scenarios, the participant observed the vehicle either rerouting around another vehicle or rerouting around a geofenced area. For Safe2Ditch scenarios, the participant was instructed to initiate an automated emergency landing (i.e., ditch) and observe the reroute and landing. The GCSOs were also required to coordinate departure times and observe simulated traffic using a prototype research software on a separate display from MPATH (not discussed in this paper). A small subset of these traffic vehicles also appeared in MPATH, specifically the vehicles that would cause conflicts with the GCSO's vehicle. Tasks associated with MPATH were the same across both activities; however, the EVLOS flight activity included some additional tasks beyond the scope of MPATH, including completion of a more extensive preflight checklist, communication with a safety pilot in the field, and coordination with additional personnel in ROAM. Importantly, each GCSO was only controlling one vehicle during flight operations in both flight activities. Additionally, during these flight operations, the automation transparency features in MPATH were only used in a limited capacity due to configuration limitations on board the vehicle that prevented the transmission of data from ICAROUS and Safe2Ditch. This limited access to automation-related information is important to consider with respect to the usability results.

At the end of the full study session, each participant completed a usability questionnaire [22] for the MPATH GCS. This questionnaire was included in both the simulated and real (i.e., EVLOS) flight activities. Additionally, participants were able to provide general feedback via an open-response question after each scenario. Researchers gathered additional feedback and observations related to the MPATH GCS throughout the duration of the studies. Participants did not receive any direct benefits for participating in these studies. This research complied with the American Psychological Association Code of Ethics and was approved by NASA's Institutional Review Board. Informed consent was obtained from each participant.

B. Results and Discussion

1. *Usability Questionnaire*

Subjective MPATH usability ratings were collected from participants using the Post-Study System Usability Questionnaire (PSSUQ) [22]. This questionnaire consists of 18 questions divided into the following three categories: usability, information quality, and interface quality. Each question is scored on a scale from 1 (*strongly disagree*) to 7 (*strongly agree*). Figures 3, 4, 5, and 6 show the mean ratings and standard deviations for each category and each question from the simulated flight activity ("Flight Activity #1") and the EVLOS flight activity ("Flight Activity #2"). The category ratings were calculated by averaging the scores of the questions contained within each category. The results from this questionnaire are presented as descriptive statistics that provide clues to understanding where usability issues may be hiding. Thus, differences discussed and depicted in the figures do not imply statistical significance and are not meant to generalize beyond these participants. Rather, the results indicate high-level trends that may warrant additional investigation using other sources of data (e.g., user feedback). The usability results from the two flight activities are depicted separately in the figures but are discussed collectively since the overall trends remained relatively consistent across the activities.

Overall, the MPATH GCS was rated high for usability and interface quality but slightly lower for information quality (see Fig. 3). However, the mean rating for information quality was still relatively high (i.e., above the midpoint) for both flight activities. These results imply that users generally felt comfortable using MPATH and generally liked the user interface. Yet there may be some underlying issues with the availability of information. To better understand these potential issues, we can evaluate the results of the individual questions within the information quality category.

The information quality results (see Fig. 5) indicate two questions with mean ratings below the midpoint, whereas the other five questions in the category have relatively high ratings (i.e., above the midpoint). The first low-rated question implies that MPATH did not give error messages that clearly indicated how to fix problems, and the second low-rated question implies that on-screen messages provided in MPATH were not clear. Both of these findings point to issues with alerts and messages that are further explored using participant feedback data in the next section.

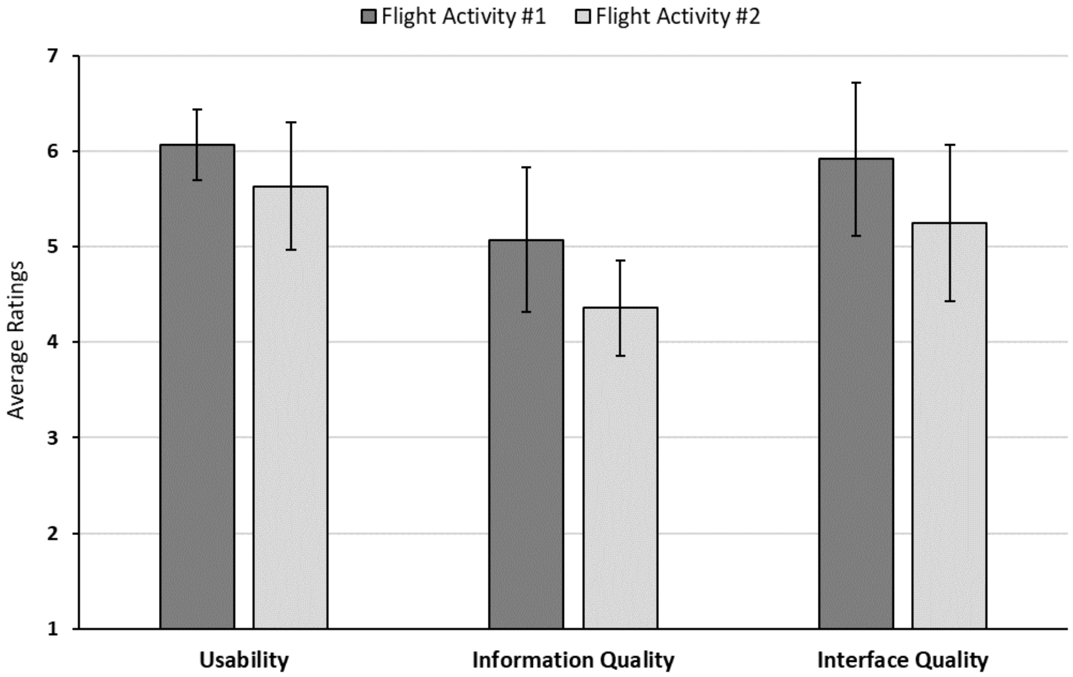


Fig. 3 Average PSSUQ ratings by category for MPATH. ***Note*****: Error bars represent standard deviations.**

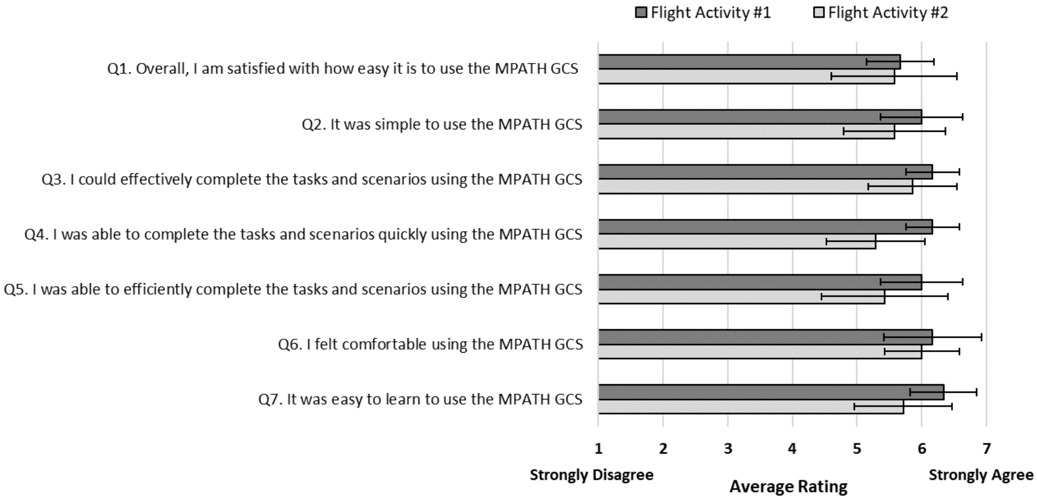


Fig. 4 Average usability item ratings for MPATH GCS. ***Note*****: Error bars represent standard deviations.**

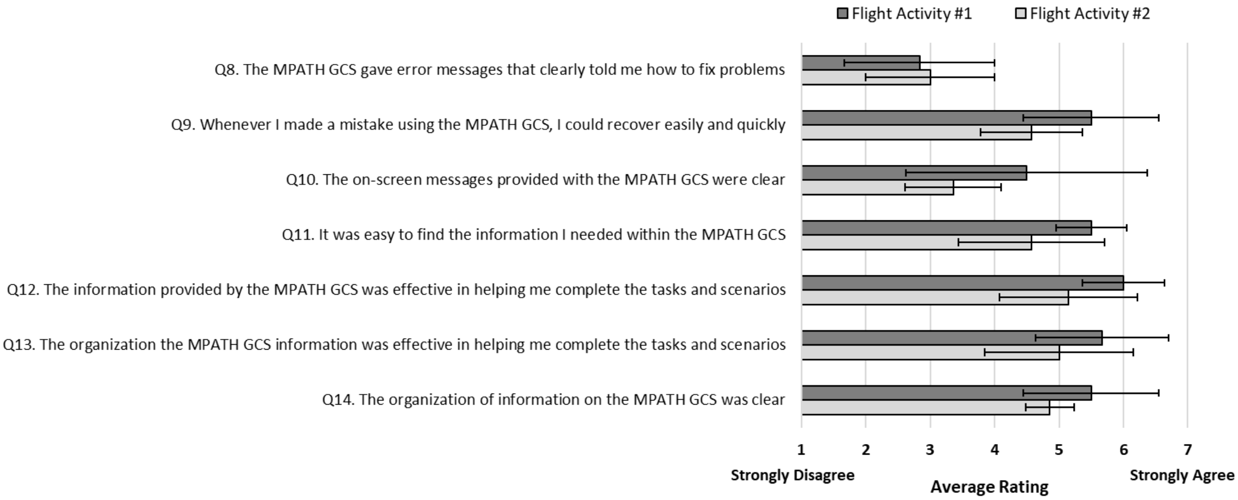


Fig. 5 Average information quality item ratings for MPATH GCS. *Note*: Error bars represent standard deviations.

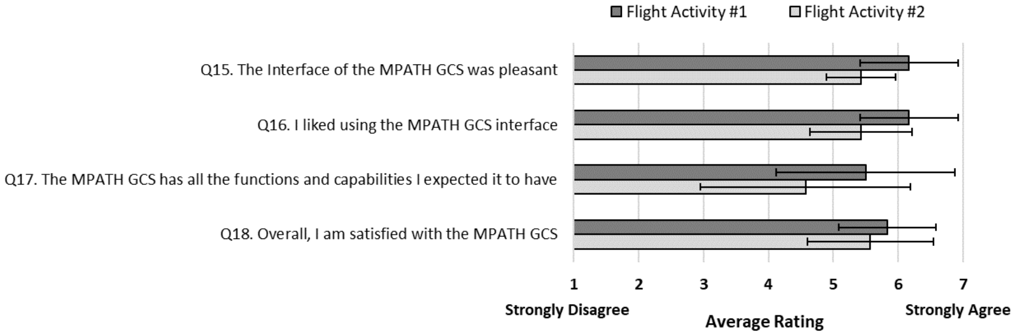


Fig. 6 Average interface quality item ratings for MPATH GCS. *Note*: Error bars represent standard deviations.

2. *General Feedback*

Subjective feedback for the MPATH GCS was collected across both flight activities from the following four sources: open-response question (post-flight), open-response question for each PSSUQ rating (post-study), short interviews (i.e., conversations; during study), and researcher observations (during study). Data from these four sources were filtered to include only MPATH-related comments and exclude bug-related issues (i.e., not generalizable). Positive feedback was also excluded as the purpose of this paper is to discover opportunities for improvements. The responses were then categorized by general themes, which we established as potentially useful groupings for both developers and researchers. This general feedback for MPATH is presented in Table 1. Whereas the data included here span only the sources described, additional follow-up discussions were conducted with participants to explore issues related to the flight operations more broadly (see Ref. [21]). That information is not included here but may provide more insight into the underlying thought processes associated with the feedback presented in this paper. Importantly, the feedback discussed in this section is for single-vehicle control from MPATH. Furthermore, because data were collected during two specific flight activities, some feedback may not generalize to other flight operations (i.e., specific feedback may be an artifact of a specific flight activity).

Notably, although the table only includes feedback targeting GCS deficiencies, the positive feedback provided by participants generally supported the findings of the usability questionnaire. For example, several GCSOs stated that the MPATH GCS interface felt familiar because of their previous experience with QGroundControl, which is reflected in the high usability category ratings presented in the previous section.

Table 1 Participant feedback for MPATH from the two initial usability studies

Primary Category	Secondary Category	Specific Feedback
Information Availability	Alerts	- Usually multiple nuisance error alerts during flight - Error alert banner can be distracting when error is not critical - Alerts should still be apparent when widgets are collapsed
	Automation Transparency	- Aircraft intended flight path not clear during auto resolution - Heading bands for collision avoidance would be helpful - Auto-selected emergency landing site not clear during ditch
	Customizability	- Prefer ability to rearrange widgets to meet individual needs
	Mode Awareness	- Did not notice GCS link was lost until post-flight - Noticed lost link but informed that this was normal - Need autopilot mode changes to be more visually apparent - Need automation active status to be more visually apparent
	Traffic Information	- Need altitude, speed, and intent of traffic aircraft on the map - Show source of traffic data if known (e.g., ADS-B, local radar) - Show last known update time for individual traffic vehicle data - Show preceding flight track paths of traffic vehicles on map - Display all traffic information in GCS rather than separate tools - Not sure what traffic data to trust when distributed across tools - Traffic icon should match vehicle type (e.g., airplane, drone)
	Vehicle / Mission State	- Important info is not placed in easy to view locations (e.g., lost link) - Remove instances of redundant data to reduce scanning and clutter - Need altitude and climb rate for own vehicle next to icon on map - Unable to monitor active flight during in-flight plan change - Had to compare lat/lon for ditch sites that should be indicated on map - Eliminate unneeded scrolling and utilize color coding for ditch sites - Geofence boundaries should be a different color than flight path - Include clock (local time) with seconds on main map display - Need confirmation of connect to / disconnect from vehicle - Display routine landing sites on map
	Vertical Position	- Need visualization of vertical positioning beyond numerical data - Include flight plan altitude profile view on main map display - Include altitude and speed when hovering on vertical profile view
Action Implementation	Action Prompts	- Confirmation prompt should be more prominent - Confirmation prompt should only appear when confirm is expected
	Flight Operation Review	- Improve interface to support flight plan reviews - Unable to interact with active flight during in-flight plan change
	Hidden / Buried Actions	- Provide easier access to common actions (e.g., vehicle connect)
	Quick Actions	- Need instant Return to Launch / failsafe action (less than 5 seconds) - Need quick response to unexpected landing denied error - Need manual control of descent rate for traffic hazards
	Unneeded Redundancy	- Remove unnecessary redundant buttons / controls

IV. Multi-Aircraft Control Flight Assessment

A. Method

In 2022, the MPATH GCS was used by a single GCSO to remotely control two aircraft simultaneously. This flight assessment employed a hybrid configuration in which one vehicle was real (EVLOS operation) and one vehicle was simulated. The flight assessment spanned several weeks and was conducted in the ROAM UAS Operations Center at NASA Langley Research Center [20] (see Fig. 2). This flight activity was not associated with a study, so there were no participants. Rather, the GCSO and flight crew were researchers, and lessons learned were collected throughout the operations.

The GCSO's duties were nearly identical to those in the initial usability study except that the GCSO did not use the prototype research software for coordinating departure times. However, the GCSO used a new prototype 3-dimensional (3D) display on a screen separate from MPATH that displayed the two vehicles and flight paths with a detailed rendering of the physical environment. The simulated vehicle was equipped with ICAROUS and Safe2Ditch, but automated resolutions were not enabled during these flights (i.e., the automation could not take control of the aircraft). The real aircraft was not equipped with ICAROUS or Safe2Ditch.

B. Lessons Learned

The lessons learned from the multi-vehicle control flight assessment are presented as researcher feedback in Table 2. This feedback is organized into the same categories as participant feedback from the initial usability studies (see Table 1). Empty cells in the table indicate categories that did not have associated feedback during this assessment. Importantly, the multi-vehicle control functionality in this version of MPATH (v1.0.2) is the same as the standard QGroundControl (v4.1.3) functionality, so these data provide a useful baseline for which to compare future MPATH modifications to multi-vehicle control features.

Although not mentioned in the table, the GCSO reported that the prototype 3D display was very useful for visualizing relative positions and altitudes of aircraft. This feedback indicates that the 3D display could be a candidate tool for addressing the issues related to vertical position information exposed during the initial usability study.

Table 2 Researcher feedback for MPATH from the multi-aircraft control flight assessment

Primary Category	Secondary Category	Specific Feedback
Information Availability	Alerts	- Need to prevent pop up alerts during process of switching vehicles - Need alerts when other personnel reach critical points in checklist
	Automation Transparency	
	Customizability	
	Mode Awareness	
	Traffic Information	
	Vehicle / Mission State	- Need ability to label vehicles manually and maybe set vehicle icons - Need more awareness of own vehicle states while not in focus
	Vertical Position	
Action Implementation	Action Prompts	- Should have to confirm current vehicle before issuing any commands
	Flight Operation Review	- Integrate shared checklists with other personnel across vehicles
	Hidden / Buried Actions	
	Quick Actions	- Need to be able to switch between vehicles with one click
	Unneeded Redundancy	

V. Design Recommendations

Based on the results of the two usability studies and the multi-vehicle control assessment, several GCS design recommendations are proposed. These recommendations attempt to avoid addressing bugs or minor changes to the MPATH interface but rather target generalizable issues that might be applicable to any designer or developer of a GCS. The recommendations described here are derived from the results presented in the previous sections, but notably, these recommendations do not capture every issue previously presented or discussed. It may be useful for designers and developers to reevaluate the data in Tables 1 and 2 to derive additional insights. The following recommendations provide general guidance for GCS development based on data collected in the ROAM UAS Operations Center at NASA Langley Research Center.

1. Anticipate Alarm Fatigue and Response Automatization

Be aware of any possibility for repetitive errors or alerts (i.e., nuisance alerts) that could lead the human operator to experience alarm fatigue [23], which may result in important information being overlooked as alerts are repeatedly dismissed. Additionally, be aware of user actions or sequences of actions that may be susceptible to automatization (i.e., reflexive or habit-formed responses [24]), and ensure that an automatic response (by the human) is not triggered erroneously in situations where the resulting action could lead to hazardous outcomes.

Examples of these issues in the context of GCS software include pressing a button to dismiss nuisance alerts and sliding a slider bar (upon appearance of the slider; see slider at the bottom of Fig. 1) to confirm an action. As a GCSO uses the GCS software, certain repeated responses may be executed without processing the information or actions associated with that response. This can be beneficial to operator performance in some cases (e.g., efficiency); however, designers and developers should make an effort to avoid overwhelming the human operator with alerts and avoid accidentally triggering responses from the human operator. For example, if an operator regularly receives nuisance alerts, they may begin dismissing all alerts without reading the alert message, which could result in the operator missing critical information. If these repetitive alerts cannot be avoided, effort should be made to protect the GCSO and the vehicle from issues associated with missed information. As an example, one idea for addressing the issue of nuisance alerts is to establish a method for muting specific error messages for the remainder of a flight (but note that this approach would likely require a quick method for unmuting error messages to avoid potential issues of missed information).

2. Provide Automation Transparency

While the human operator holds responsibility (i.e., accountability) for safety of flight, it is important that they know where the vehicle will be traveling at any given time. In situations where automation has the authority to control a vehicle, designers and developers should make an effort to provide information to the operator that depicts the automation's intent (i.e., what the automation is doing and planning to do).

For example, in the case of detect and avoid (DAA) automation (e.g., ICAROUS), when the automation takes control of the vehicle to avoid a potential collision, the GCS should display the new flight path of the aircraft or at least some information that helps the operator understand the unfolding situation and the aircraft's intended direction of travel. Another option for automation transparency with DAA automation is to provide color-coded heading bands around the vehicle that show the heading ranges that the DAA automation is actively avoiding (an approach used with the ICAROUS software). Note that the MPATH GCS software includes these features, but the functionality was not available during the usability studies because the vehicles were not properly configured.

3. Enable Quick Commands to Vehicle

While the human operator holds responsibility (i.e., accountability) for safety of flight, it is important that they have the ability to quickly respond to potentially hazardous situations. Be aware of situations where the operator may need to quickly execute an action, and ensure that those actions are readily available to the operator when needed. Familiarity of responses should not be expected for these types of situations (e.g., emergencies) as they may rarely be encountered, so the interface for executing actions should be clear and straightforward.

For example, if a vehicle unexpectedly encounters a traffic conflict, it would be useful for the operator to have the ability to quickly descend at a specified rate to avoid a collision. Generally, if quick actions are not available, operators may have limited or inadequate options for responding to critical situations.

4. Collocate Task-Related Information

To reduce visual scanning and promote efficient information processing, information associated with a common task should have high spatial proximity where possible (see proximity compatibility principle [15]).

For example, to determine a vehicle's position, the operator looks at the vehicle icon on the map (for horizontal position information) and then scans to the numerical altitude display (for vertical position information). The need to scan away from the vehicle icon for altitude information is a result of low spatial proximity and could lead to issues such as change blindness (i.e., failing to notice something is different from what it was) [25]. However, by collocating the altitude information with the vehicle icon, this repeated scanning and the issues associated with it can be avoided. An idea for addressing this issue more generally is the use of modular widgets in the display interface to collocate information related to a specific system or task, which is a feature of the MPATH GCS interface that was previously discussed.

5. Reinforce Vertical Position Information

Presenting 3D vehicle position information on a 2-dimensional (2D) map display inherently requires a decoupled representation of the horizontal position (i.e., latitude, longitude) and the vertical position (i.e., altitude). This decoupling forces the operator to mentally integrate at least two sources of information to interpret the position of the vehicle in three-dimensional space. Integration of these data sources by the operator may be less efficient if the information is presented only numerically (i.e., lacking pictorial realism) rather than with the addition of a graphical representation (e.g., Ref. [26]). Thus, to support efficient information processing and decision making, the vertical position of the aircraft and the vertical dimension of the flight path should be presented in a manner that eases integration with horizontal position information.

This could be addressed with the inclusion of a 2D profile (or side) view of the aircraft position, flight path, terrain, and obstacles. As mentioned previously, a 3D representation of the environment could also potentially be used to address this issue; however, assuming the 2D map view remains the primary display and the 3D display is supplementary, a solution for the 2D map view would likely still be necessary. Note that air traffic control (ATC) displays also face this challenge; thus, existing solutions and research in the ATC domain (e.g., Ref. [27]) may be useful to consider for GCS development.

6. Support Flight Plan Reviews

Assuming the GCSO is required to review flight plans prior to takeoff, effort should be made to streamline the flight plan review process. Additionally, if a new flight plan can be uploaded during a flight, the GCSO should have the ability to review the new plan without hiding any information related to active flights.

Flight plan reviews should be integrated into the GCS display rather than a separate display to avoid issues associated with visual scanning and divided attention (e.g., change blindness). Additionally, the flight plan preview should be integrated with the primary map display to allow monitoring of active flights during review. As a possible solution for implementing a review process, the GCS could require the operator to cycle through each mission point (i.e., include ‘Next’ and ‘Previous’ buttons) to confirm the details of each mission point before uploading to the vehicle.

7. Support Shared Checklists (Multi-Vehicle Control)

While the human operator holds responsibility (i.e., accountability) for safety of flight, it is important that they have awareness of the vehicle status at any given time. This is particularly important during operations in which multiple non-collocated personnel (e.g., ground support crew, remote GCSO) interact with the vehicle at various stages of the flight operation. To address this issue, all personnel involved in the operation should have access to a shared checklist that updates in real time as each item is marked complete.

This functionality would enable the GCSO to anticipate upcoming tasks (e.g., initiate takeoff procedures upon completion of vehicle inspection by ground support crew) and maintain vehicle status awareness without having to track progress manually or in working memory. This is especially important for a single GCSO who is responsible for managing more than one vehicle. Integrating this shared checklist within the GCS software (as opposed to on a separate display) could facilitate better GCSO performance by reducing visual scanning, but importantly, as more elements are added to the GCS display, appropriate design decisions should be made to alleviate display clutter. Note that this issue was observed for multi-vehicle control, but the proposed functionality could also be useful for single-vehicle control operations.

8. Streamline Vehicle Switching (Multi-Vehicle Control)

To support multi-vehicle control by a single GCSO, effort should be made to facilitate the following software behaviors: one-click action to change the current active (i.e., in focus) vehicle, clear distinction of the active vehicle on the display, and required confirmation of the current active vehicle before issuing any vehicle commands.

If the operator encounters additional steps (e.g., pop-up alerts) when changing active vehicles, they may have trouble responding quickly to potentially hazardous situations. If there is any ambiguity regarding the currently active vehicle, the operator may mistakenly attribute vehicle information or issue a command to a wrong vehicle, which could lead to hazardous situations. To reduce the likelihood of ambiguity and confusion, the current active vehicle should be clearly differentiated from the non-active vehicles (e.g., more salient vehicle icon).

VI. Conclusion

The MPATH GCS software has demonstrated to be a useful capability for human factors research evaluating remote operators of AAM vehicles. The two initial usability studies and the multi-vehicle control assessment generated valuable feedback, which was used to derive generalized recommendations for GCS software design and development. Generally, participants rated MPATH high on usability and interface quality, whereas information quality was rated slightly lower. These results were supported by specific feedback from open-response questions, short interviews, and researcher observations. Future updates to the MPATH GCS software will incorporate the recommendations that have been discussed. Additionally, future updates will include improvements to the multi-vehicle control interface.

References

[1] Federal Aviation Administration, "UTM Concept of Operations Version 2.0 (UTM ConOps v2.0)," Mar. 2020. [Online]. Available: https://www.faa.gov/uas/research_development/traffic_management/

[2] Federal Aviation Administration, "UAM Concept of Operations Version 1.0 (UAM ConOps v1.0)," Jun. 2020. [Online]. Available: https://nari.arc.nasa.gov/sites/default/files/attachments/UAM_ConOps_v1.0.pdf

[3] K. Antcliff *et al.*, "Regional Air Mobility: Leveraging Our National Investments to Energize the American Travel Experience," *NASA NTRS*, 2021.

[4] S. A. Mouloua, J. Ferraro, M. Mouloua, and P. A. Hancock, "Trend Analysis of Unmanned Aerial Vehicles (UAVs) Research Published in the HFES Proceedings," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 62, no. 1, pp. 1067–1071, Sep. 2018, doi: 10.1177/1541931218621245.

[5] E. T. Chancey and M. S. Politowicz, "Public Trust and Acceptance for Concepts of Remotely Operated Urban Air Mobility Transportation," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 64, no. 1, pp. 1044–1048, Dec. 2020, doi: 10.1177/1071181320641251.

[6] N. Rymer, A. Moore, S. Young, L. Glaab, K. Smalling, and M. Consiglio, "Demonstration of Two Extended Visual Line of Sight Methods for Urban UAV Operations," in *AIAA AVIATION 2020 FORUM*, VIRTUAL EVENT, Jun. 2020. doi: 10.2514/6.2020-2889.

[7] M. S. Politowicz, E. T. Chancey, and L. J. Glaab, "Effects of Autonomous sUAS Separation Methods on Subjective Workload, Situation Awareness, and Trust," in *AIAA Scitech 2021 Forum*, VIRTUAL EVENT, Jan. 2021. doi: 10.2514/6.2021-1176.

[8] M. D. Patterson *et al.*, "An Initial Concept for Intermediate-State, Passenger-Carrying Urban Air Mobility Operations," in *AIAA Scitech 2021 Forum*, VIRTUAL EVENT, 2021. doi: 10.2514/6.2021-1626.

[9] V. V. Aubuchon, K. E. Hashemi, R. J. Shively, and J. M. Wishart, "Multi-Vehicle (m:N) Operations in the NAS - NASA's Research Plans," in *AIAA AVIATION 2022 Forum*, Chicago, IL & Virtual, Jun. 2022. doi: 10.2514/6.2022-3758.

[10] T. Sato, M. S. Politowicz, S. Islam, E. T. Chancey, and Y. Yamani, "Attentional Considerations in Advanced Air Mobility Operations: Control, Manage, or Assist?," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 66, no. 1, pp. 28–32, Sep. 2022, doi: 10.1177/1071181322661184.

[11] E. T. Chancey, M. S. Politowicz, and L. Le Vie, "Enabling Advanced Air Mobility Operations through Appropriate Trust in Human-Autonomy Teaming: Foundational Research Approaches and Applications," in *AIAA Scitech 2021 Forum*, VIRTUAL EVENT, Jan. 2021. doi: 10.2514/6.2021-0880.

[12] E. T. Chancey *et al.*, "Foundational Human-Autonomy Teaming Research and Development in Scalable Remotely Operated Advanced Air Mobility Operations: Research Model and Initial Work," in *AIAA Scitech 2023 Forum*, National Harbor, MD, 2023.

[13] Dronecode Project, “MAVLink (Micro Air Vehicle Communication Protocol).” Dronecode Project, Inc., 2021. [Messaging protocol]. Available: https://mavlink.io/en/

[14] Dronecode Project, “QGroundControl.” Dronecode Project, Inc., 2019. [Computer software]. Available: http://qgroundcontrol.com

[15] C. D. Wickens and C. M. Carswell, “The Proximity Compatibility Principle: Its Psychological Foundation and Relevance to Display Design,” *Human Factors*, vol. 37, no. 3, pp. 473–494, Sep. 1995, doi: 10.1518/001872095779049408.

[16] M. Consiglio, C. Munoz, G. Hagen, A. Narkawicz, and S. Balachandran, “ICAROUS: Integrated Configurable Algorithms for Reliable Operations of Unmanned Systems,” in *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, Sacramento, CA, USA, Sep. 2016, pp. 1–5. doi: 10.1109/DASC.2016.7778033.

[17] L. J. Glaab *et al.*, “Safe2Ditch Autonomous Crash Management System for Small Unmanned Aerial Systems: Concept Definition and Flight Test Results,” *NASA NTRS*, 2018, [Online]. Available: https://ntrs.nasa.gov/api/citations/20180008439/downloads/20180008439.pdf

[18] J. B. Lyons, "Being transparent about transparency: A model for human-robot interaction," in *Trust and Autonomous Systems: Papers from the 2013 AAAI Spring Symposium*, 2013.

[19] J. R. Unverricht, E. T. Chancey, M. S. Politowicz, B. K. Buck, and S. C. Geuther, “Eye Glance Behaviors of Ground Control Station Operators in a Simulated Urban Air Mobility Environment,” in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, Portsmouth, VA, USA, Sep. 2022, pp. 1–6. doi: 10.1109/DASC55683.2022.9925872.

[20] B. K. Buck, E. T. Chancey, M. S. Politowicz, J. R. Unverricht, and S. C. Geuther, “A Remote Vehicle Operations Center’s Role in Collecting Human Factors Data,” in *AIAA Scitech 2023 Forum*, National Harbor, MD, 2023.

[21] J. R. Unverricht, E. T. Chancey, M. S. Politowicz, B. K. Buck, S. C. Geuther, and K. M. Ballard, “Where is the Human in the Loop? Analysis of Extended Visual Line of Sight Uncrewed Aerial System Operations within a Remote Operations Environment,” in *AIAA Scitech 2023 Forum*, National Harbor, MD, 2023.

[22] J. R. Lewis, “Psychometric Evaluation of the Post-Study System Usability Questionnaire: The PSSUQ,” *Proceedings of the Human Factors Society Annual Meeting*, vol. 36, no. 16, pp. 1259–1260, Oct. 1992, doi: 10.1177/154193129203601617.

[23] J. M. Solet and P. R. Barach, “Managing Alarm Fatigue in Cardiac Care,” *Progress in Pediatric Cardiology*, vol. 33, no. 1, pp. 85–90, Jan. 2012, doi: 10.1016/j.ppedcard.2011.12.014.

[24] I. Barshi and A. F. Healy, “Checklist Procedures and the Cost of Automaticity,” *Memory & Cognition*, vol. 21, no. 4, pp. 496–505, Jul. 1993, doi: 10.3758/BF03197181.

[25] R. A. Rensink, “Change Detection,” *Annual Review of Psychology*, vol. 53, no. 1, pp. 245–277, Feb. 2002, doi: 10.1146/annurev.psych.53.100901.135125.

[26] D. G. Morrow, C. M. Hier, W. E. Menard, and V. O. Leirer, “Icons Improve Older and Younger Adults’ Comprehension of Medication Information,” *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, vol. 53B, no. 4, pp. P240–P254, Jul. 1998, doi: 10.1093/geronb/53B.4.P240.

[27] E. M. Palmer, T. C. Clausner, and P. J. Kellman, “Enhancing Air Traffic Displays via Perceptual Cues,” *ACM Transactions on Applied Perception*, vol. 5, no. 1, pp. 1–22, Jan. 2008, doi: 10.1145/1279640.1279644.