Using Mixed Reality and Artificial Intelligence for Complex Task Guidance in a UH-60 Environment

Brian Williamson, Pierce Powell, Ryan Ghamandi, Jacob Belga, Joseph J. LaViola Jr.
University of Central Florida
Orlando, Florida
Brian.Williamson@ucf.edu, Pierce.Powell@ucf.edu, Ryan.Ghamandi@ucf.edu, Jacob.Belga@ucf.edu, JLaviola@ucf.edu

Nayan N. Chawla, Ryan P. McMahan
Virginia Tech
Blacksburg, Virginia
nnchawla@vt.edu, rpm@vt.edu

Michael Middleton, Molly Kluck, Ryan McKendrick
HAPII Lab, Northrop Grumman
Falls Church, Virginia
Michael.Middleton@ngc.com, Molly.Kluck@ngc.com, Ryan.McKendrick@ngc.com

ABSTRACT

Military personnel face increasingly complex tasks and systems; however, human training and monitoring has had much less development. Often these complex systems require significant, costly time and training for knowledge transfer from experts to trainees. Additionally, experts in these complex systems can become unengaged and complacent when performing repetitive tasks, which can lead to catastrophic errors. By utilizing advancements in mixed reality and artificial intelligence, an effective guidance system with a modern interface can be developed for training and task monitoring. In this paper, we detail a task guidance system designed to monitor and provide guidance to pilots as they go through simulated pre-flight procedures and in-flight emergency procedures within a UH-60 helicopter. We further introduce a low cost, simulator environment for the UH-60 helicopter that both projects the cockpit alongside the outside environment and provides haptic feedback through a collocated touchscreen interface. This environment is built on top of insights gained from real pilot data we collected in a multi-million dollar UH-60 flight simulator. The low-cost environment had pilots go through a 250-step preflight procedure and an in-flight scenario that could trigger any combination of 3 emergencies: a heading mis-compare, a single engine failure, and an inclement weather event. We translated expert knowledge and procedural documents into a knowledge representation for an artificial intelligence, which used mixed reality to guide users via look and operational cues. We examine the use of Pick Arrow, Ghost Hand, and Target Outline operational cues; as well as Look Arrow, Attention Funnel, and EyeSee360 look cues. We test both novices with full guidance and experts with guidance only when they have been predicted to need it. Our experiments show the capability for mixed reality and artificial intelligence to provide effective training and error monitoring while reducing cost and time requirements.

ABOUT THE AUTHORS

Brian M. Williamson is a faculty researcher at the University of Central Florida in the department of Electrical Engineering and Computer Science within the Interactive Computing Experiences Research Cluster under Dr. LaViola. His primary research includes 3D user interfaces with previous papers written on the RealNav system and AgileSLAM algorithm.

Pierce Powell is a third-year Ph.D. student in Computer Science at the University of Central Florida, where he conducts research under Dr. Johnathan Mell in the SCION-AI Lab. His work explores the impact of unconventional techniques in virtual reality—such as impossible spaces, motion cues, and disembodiment—on user experience and presence. He has published on evolutionary computation at ICMLA 2023 and is currently applying his AI/ML expertise as an Engineering Intern at SoarTech.

Ryan Ghamandi is a Ph.D. Candidate at the University of Central Florida specializing in the research and development of XR Collaboration and Natural Multimodal User Interfaces in the Interactive Computing Experiences Research Cluster under Dr. LaViola.

Jacob Belga is a Ph.D. Candidate at the University of Central Florida specializing in developing and designing virtual environments for VR experiences in the Interactive Computing Experiences Research Cluster under Dr. LaViola. His main area of research is how to leverage the technological features of an environment to influence sense of presence.

His most recent paper, *The Fidelity-based Presence Scale (FPS): Modeling the Effects of Fidelity on Sense of Presence,* won Best Paper at the 2025 CHI conference.

Dr. Joseph J. LaViola Jr. is the Charles N. Millican Professor of Computer Science and directs the Interactive Computing Experiences Research Cluster at the University of Central Florida. He is the former director of the Modeling and Simulation graduate program at UCF and was an adjunct research professor and visiting scholar at Brown University. His primary research interests include pen- and touch-based interactive computing, virtual and augmented reality, 3D spatial interfaces, human-robot interaction, multimodal interaction, and user interface evaluation. He has published over 200 refereed journal and conference papers, 8 book chapters, and has 5 patents. His work has appeared in journals such as ACM TIIS, ACM TOCHI, IEEE PAMI, Presence, and IEEE Computer Graphics & Applications, and he has presented research at conferences including ACM CHI, ACM IUI, IEEE Virtual Reality, and ACM SIGGRAPH. He is also the lead author on the second edition of "3D User Interfaces: Theory and Practice", the first comprehensive book on 3D user interfaces. In 2009, he won an NSF Career Award to conduct research on mathematical sketching. In 2025, he was inducted into the IEEE VGTC Virtual Reality Academy. Joseph received a Sc.M. in Computer Science in 2000, a Sc.M. in Applied Mathematics in 2001, and a Ph.D. in Computer Science in 2005 from Brown University. He is a senior member of the ACM and IEEE.

Nayan N. Chawla is a Ph.D. Student at Virginia Polytechnic Institute and State University specializing in machine learning prediction on virtual reality data as a member of the Extended Reality and Artificial Intelligence (Xrai) Lab under Dr. Ryan McMahan. His main area of research focuses on using machine learning models to predict real world transfer of competence from virtual contexts to real world scenarios.

Dr. Ryan P. McMahan is the Director of the Center for Human-Computer Interaction (CHCI) at Virginia Tech (VT) and a Professor of Computer Science. He also directs the Extended Reality & Artificial Intelligence (Xrai) Lab at VT. He has best paper and honorable mention awards from ACM CHI and IEEE VR, respectively.

Michael J. Middleton is an applied AI researcher at Northrop Grumman and PhD student at CU Boulder under Dr. Leanne Hirschfield. His research interests are in procedural content generation, neuroadaptive AI, and generative planning.

Molly Ann Kluck is a human-machine teaming scientist at Northrop Grumman with a focus on applying human factors, methods and cognitive science to autonomy in order to make it safe, effective, and fun.

Dr. Ryan McKendrick is a Senior Staff Cognitive Scientist and Technical Fellow at Northrop Grumman Mission Systems. He received a Ph.D. from George Mason University in Human Factors and Applied Cognition in 2016. He also holds an active TS/SCI clearance. Dr. McKendrick has a number of publications exploring the neuroergonomics of mobile displays, the interaction between distractions, mobility and cognitive work, the neurovascular effects of noninvasive brain stimulation, as well as cognitive load classification. His current work focuses on real-time neuro-adaptive systems for human-machine teaming, neuro-cognitive profiling, AI dynamics discovery and multi-objective decision making mission planning and optimization. Dr. McKendrick actively consults within Northrop Grumman on multiple S&T programs and proposals and has been the principal investigator (PI) on multiple DARPA programs and is the current PI of DARPA AIR.

Using Mixed Reality and Artificial Intelligence for Complex Task Guidance in a UH-60 Environment

Brian Williamson, Pierce Powell, Ryan Ghamandi, Jacob Belga, Joseph J. LaViola
University of Central Florida
Orlando, Florida
Brian.Williamson@ucf.edu, Pierce.Powell@ucf.edu, Ryan.Ghamandi@ucf.edu, Jacob.Belga@ucf.edu, JLaviola@ucf.edu

Nayan Chawla, Ryan McMahan
Virginia Tech
Blacksburg, Virginia
Nayan.Chawla@vt.edu, rpm@vt.edu

Michael Middleton, Molly Kluck, Ryan McKendrick
HAPII Lab, Northrop
Falls Church, Virginia
Michael.Middleton@ngc.com, Molly.Kluck@ngc.com, Ryan.McKendrick@ngc.com

INTRODUCTION

Military personnel are required to undertake complex tasks in unique environments and work conditions (Kryskow, Beidleman, Fulco, & Muza, 2013) which are growing more complex over time (Socha, et al., 2020). This requires extensive knowledge transfer from experts to novice personnel in costly training scenarios (Cao, Kearns, Niechwiej-Szwedo, & Irving, 2024). However, even with appropriate time and resources made available, complications can arise on the efficacy of knowledge transfer. The passion of the expert in the field and their desire to effectively transfer knowledge can come into play (Sie, 2009) along with the need for novel tasks requiring realistic adaptive exercises for increased effectiveness (Kimball & Holyoak, 2000). Furthermore, experts can become complacent when performing repetitive tasking which can lead to severe consequences when multiple errors start to occur (Barter, Clayton, & Clark, 1993).

Thankfully advancements in artificial intelligence (AI) and mixed reality (MR) present a new opportunity to provide standardized task guidance to novice and expert operators as both a step to the knowledge transfer process and as a passive error monitoring system. This allows for low-cost effective monitoring as the AI can provide expert level guidance while the MR gives full immersion into the task.

In this paper we present a low-cost dynamic guidance simulator that utilizes both MR and AI for UH-60 Blackhawk pilots in training and expert pilot monitoring. The system is composed of two major components: the front-end system (shown in Figure 1) which focuses on mixed reality, haptic feedback, visual cues, and realistic simulations; and the back-end system (shown in Figure 2), an artificial intelligence capable of monitoring the user's actions to provide dynamic feedback directly to the front-end system. This setup allowed for a use-case specific front-end with a transferable back-end that can be easily reconfigured and used for other guidance systems.

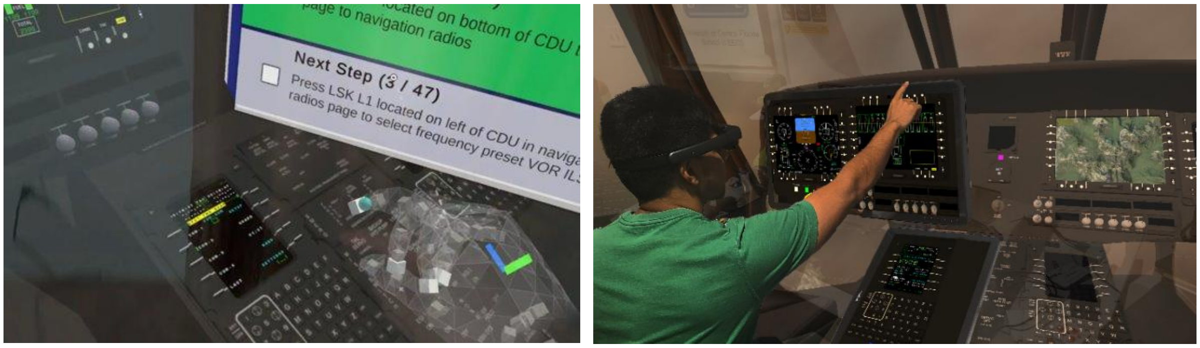

located on bottom of CDU
page to navigation radios
Next Step (3 / 47)
Press LSK L1 located on left of CDU in naviga
radios page to select frequency preset VOR IL

Figure 1. Augmented reality representation of the user's hand interacting with the virtual UH-60 console on the left. Proof of concept illustration of a user immersed in the virtual world while interacting with a touchscreen on the right.

For the front-end system we used a Magic Leap 2 MR headset with touchscreens to provide the haptic feedback of cockpit interactions. While the initial system worked with a physical cockpit, by projecting the virtual cockpit over the physical cockpit, we found that the touchscreens were sufficient to use and could dramatically reduce costs of the simulation. The back-end system ran on a remote container and the Robot Operating System (ROS) was used for communication between the two systems. This system utilized a goal-oriented action planning artificial intelligence (Orkin, 2006) for monitoring and task guidance alongside a large language model (LLM) for adapting guidance to specific user models.

We created a scenario that focuses on two unique forms of complex tasks for co-pilots in the UH-60. Pilots first had to go through a realistic multi-stage pre-flight checklist. Upon completion, the helicopter would begin flying through the simulation. Upon flight pilots would then undergo multiple emergency situations and were required to carry out emergency procedures within a limited amount of time to complete the task successfully. Failure to complete the emergency procedure would result in a crash. The first task showed a traditional transfer of knowledge in a routine procedure where the pilots had as much time as they needed. It also gave pilots an opportunity to familiarize themselves with the system. The second set of tasks was a more dynamic procedure that tested user's ability to react to an emergency.

Pilots were run through both procedures where we looked at the effectiveness of the system in teaching them to carry out the tasks. We ran a mixture of expert and novice users; expert user studies focused on error monitoring and correction while novice user studies focused on successful completion of the series and cue preference information. The initial run of our studies were too limited to fully test knowledge transfer of expertise; however, we show the effectiveness of an AI/MR system for complex task guidance in the ability for true novices to complete the complex tasks as though guided by an expert. We also lay out a framework for future work that can demonstrate knowledge transfer over time and with several runs through the simulation.

In the next section we review work related to task guidance, mixed reality, and artificial intelligence. Section three goes through the details of the front-end and back-end systems used for the simulation along with the design decision rationale. Section four discusses the studies that were conducted with the system along with system demonstrations that were performed as a part of the project. Section five concludes our discussion on the system and lays out a framework for future work.

RELATED WORK

Analysis of complex task guidance and how it can be made more efficient has a rich history both in military and civilian applications. In (Ockerman & Pritchett, 1998) wearable computers were used as a preliminary investigation of their efficacy in performing aircraft inspections. (Schutte & Trujillo, 1996) looked at the requirements of task guidance and retention for non-normal flight situations. (Degani & Wiener, 1993) went through the efficacy of flight checklists and offered suggestions of improvements to this physical system. In (Uhrmann, Strenzke, & Schulte, 2010) an evaluation was made of multi-UAV guidance by helicopter pilots with task information loaded into the UAV systems. (Kimball & Holyoak, 2000) and (Spector, 2008) looked at the need for expertise in dynamic training tasks, demonstrating the need for experts to be present during training of complex scenarios to provide adequate knowledge transfer. Steps have been made to automate this form of knowledge transfer, such as through intelligent tutoring as seen in the GIFT system (Sottilare, 2013). Preliminary efforts have also been made using early artificial intelligence, as seen in the PAL-CPOF program (Garvey, et al., 2009) and the development of the Task Assistant for military environments (Peintner, Dinger, Rodriguez, & Myers, 2009).

Similarly, there have been research efforts on the efficacy of mixed reality in military applications. In (Knerr, 2006) the efficacy of virtual simulations versus their cost was analyzed for dismounted soldier training. (Hughes, Stapleton, Hughes, & Smith, 2005) looked at the benefits of mixed reality in a variety of applications, including the use of mixed reality in Military Operations in Urban Terrain (MOUT). (LaViola, et al., 2015) looked at the use of augmented reality in normal military operations and the use of SLAM algorithms for mapping unknown environments in real-time. (Schaffernak, et al., 2022) developed novel use cases of mixed reality in pilot training, including knowledge of flight management systems for emergency scenarios and procedural training of checklists. (Macchiarella, Yu, Liu, & Vincenzi, 2023) looked at the efficacy of mixed reality in various aviation tasks, determining it to be an effective means of training for psychomotor functions and knowledge transfer of complex tasks. (King, Carmody, & Deaton, 2023) looked at the effect mixed reality training is having on the aviation industry, in particular its widespread use within flight schools, but lack of use in government certifications.

Recent advancements in artificial intelligence, particularly in large language models and knowledge representation, have created new avenues of research in complex task guidance. (McGowan, et al., 2025) introduces an end-to-end system for medical and aviation MR task guidance that uses multimodal perception (audio and video) to track objects in a 3D world space and provide AI reasoning and task-guidance over those objects. In (Wu, et al., 2024) a large language model was used to simplify text instructions so that they could take up minimal amounts of area within a mixed reality system. (Shervedani, Walter, & Zefran, 2025) looked at having an LLM and human work together to develop specific task plans for robotic guidance. There has also been research adapting these systems to the aviation realm. AviationGPT (Wang, Chou, Tien, Zhou, & Baumgartner, 2024) fine-tuned existing large language models on aviation documents. LeRAAT (Schlichting, et al., 2025) was developed to provide advice from a LLM to users within a flight simulator during emergency scenarios by using a retrieval-augmented generation (RAG) system on FAA rules.

Our research aims to take these recent advancements in mixed reality and artificial intelligence and create an interactive expert AI that can provide complex real-time task guidance from within a simulation.

SYSTEM DESIGN

We divided our system into two components. The front-end system focused primarily on the display of information in a mixed reality environment, haptic feedback, and realism of the scenario and the cockpit. The back-end system utilized containers running on a remote system for the artificial intelligence that provided world state tracking, knowledge encoding, and adaptive reasoning. This was composed of two AI systems. The primary AI would monitor the state of the system and the user's actions to provide dynamic task guidance. The secondary AI was a large-language model that would generate natural-language text that would be read to the user through a text-to-audio system. The system design is shown in Figure 2.

Our total cost of the final iteration of the simulation is available in Table 1. At less than $10k, this cost is significantly lower than the estimated cost for a physical UH-60 cockpit simulator.

Table 1. Material cost of UH-60 simulation

Item	Cost
Magic Leap 2	$3,299
Touchscreens x2	$650
Touchscreen Laptop	$1,200
Backend Laptop	$2,000
Total	$7,149

To evaluate the human using the system we additionally used a functional near-infrared spectroscopy (fNIRS) neuroimaging device. The specific fNIRS system we use is the Biopac 2000s which costs roughly $20k. This allows us to measure hemoglobin in the prefrontal cortex used with brain activity as well as physiological measures of respiration and heartrate. To assess workload, we use workload models that have been validated for both cross-task and cross-participant (McKendrick, 2019).

UH-60 Pilot Pre-evaluation

Before migrating to a purely virtual trainer, we conducted an extensive baseline study on a physical UH-60 cockpit simulator. Seven participants participated in the study: three rated rotary-wing pilots, three cockpit-software engineers, and one aviation researcher. Participants completed a combined twenty-eight simulated flights from thirty-three possible flight scenarios that yielded ~3TB of multimodal data spanning mission-computer logs, egocentric video, gaze, IMU, hand tracking, audio, and raw + classified fNIRS workload streams. All messages were time-synchronized and exported to SQLite to facilitate downstream analysis and to provide ground-truth state for our AI monitoring modules.

(Nadri, et al., 2024) used the physical cockpit dataset to construct GOMS/Cogulator models of every pre-flight and in-flight monitoring step, validating the models against the recorded button-press timelines and egocentric videos. The analysis identified three pre-flight subtasks: Check Avionics (Task 3), FMS Initialization (Task 4), and MFD/FD/FMS Setup (Task 7)-as the largest contributors to working-memory load (>= 4.4 chunks on average), while Critical Data

Mis-compare (Emer-1), Pop-up Weather (Emer-4), and Engine/Fuel Emergencies (Emer-5) were the dominant high-load events during flight. The study recommended specific emergencies to focus on our virtual implementation as well as where to attend our adaptive guidance cues with more information.

(Castelo, et al., 2024) introduced HuBar, an interactive visual analytics environment that layers fNIRS-derived workload classifications over procedure timelines, error logs and gaze/IMU traces. HuBar revealed that workload spikes consistently preceded omission or order errors in the same three high-load pre-flight subtasks flagged by the CPM study, and that novices who resolved Pop-up-Weather events with fewer gaze shifts showed lower peak workload and error counts. These findings informed of the virtual system design choices in potential modality adaptations.

In short, the UH-60 pilot pre-evaluation allowed us to implement design rules that could benefit the real UH-60 scenario and supplied us with a dataset to compare against.

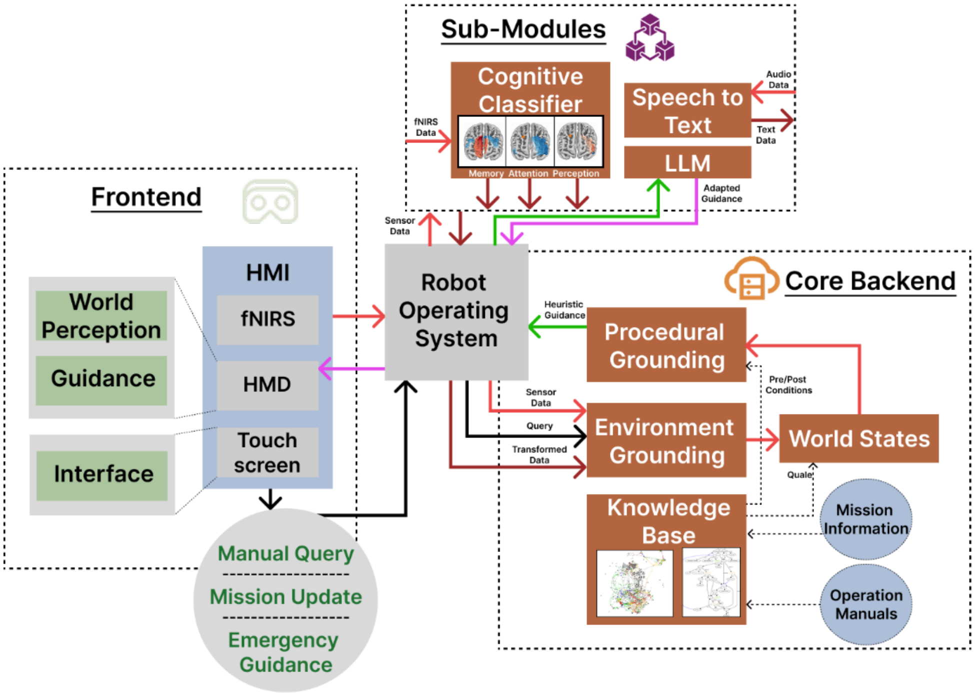


Figure 2. System design showing message flow from font-end system to back-end system and submodules

Front-end System

The front-end system utilized a Magic Leap 2 containing a build of the simulation application. The simulation was developed in Unity 3D, a popular game engine that is easily used for prototyping systems and working in mixed reality. We used ViewSonic touchscreens that provided haptic feedback for portions of the cockpit that pilots would be interacting with.

We also designed a look cue system that would create cues based on messages from the back-end system. For example, if a message came in stating the user needs to press the ENT button on the cockpit keyboard the simulation would then generate an operational cue based on its configuration, such as a floating “ghost hand”, above the button the user is supposed to press. If a message came in that the user needed to look at a particular value on their screen, the front-end system would generate a look cue, such as the look arrow and an outline to indicate where the user needed to look. The front-end system also provided periodic state updates of what buttons the user has pressed to the back-end system so that the back-end knows how to respond to different user actions given active procedures. For novice pilots, these cues were always displayed and for experts these cues were hidden until an error was detected or predicted to be made.

We used visual cutouts to create a mixed reality simulation rather than a fully VR scene. The Magic Leap 2 has a semi-transparent glass screen that could either become opaque for displaying VR scenes or transparent for viewing the real world. Furthermore, the system can control the glass in segments, allowing sections of the real-world to be seen while the rest would display virtual reality. To utilize this system, we created 3D objects in Unity with a special blackout texture applied that would signal the Magic Leap 2 to use the transparent glass for those sections. This allowed us to cut out segments of the VR scene and show the real-world interfaces to the user through them.

In the initial system the entire simulation was virtual, shown in Figure 4. However, the lack of haptic feedback when pilots press buttons created several issues and a disconnect with the simulation. Our second iteration used the cutout system described above with a realistic representation of the UH-60 cockpit designed by Northrop Grumman. This cockpit would register actual button presses and report them back to the back-end system. This iteration of the system, shown in Figure 3, was aligned with visual markers placed onto the cockpit and the Vuforia computer vision system. Once aligned, virtual buttons that were not visible were in the same position within the headset as their real-world counterparts. The system then placed visual cues over the non-visible virtual button, and it would appear as though they were pointing to the real-world button within the headset. While this system was extremely effective, movement and testing of the physical cockpit presented problems both in resource management and cost. As such we decided to test a third system that utilized touchscreens in place of the realistic cockpit, shown in Figure 4. These touchscreens were low-cost and readily made available to every test team. While there was a loss in realism, this was mitigated with better visual cues of the button being pressed on the touchscreen, and enhanced audio cues in the headset of the button press. Initial prototypes showed pilots were still immersed within the system.

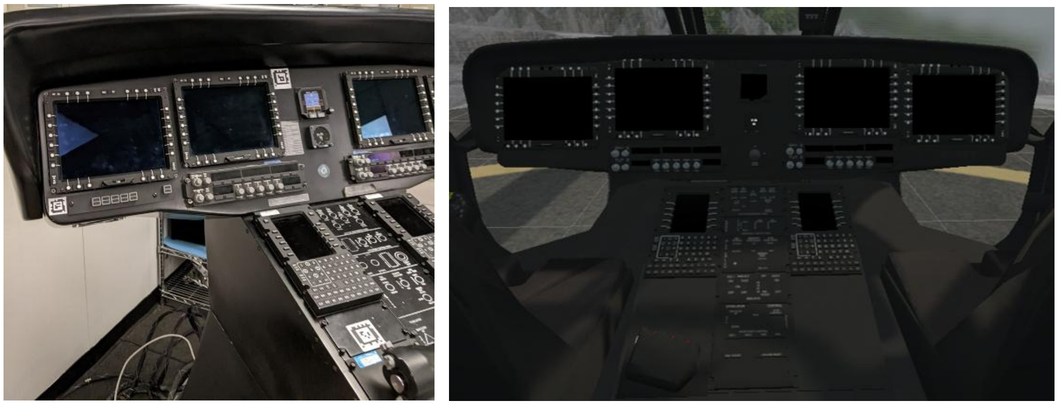
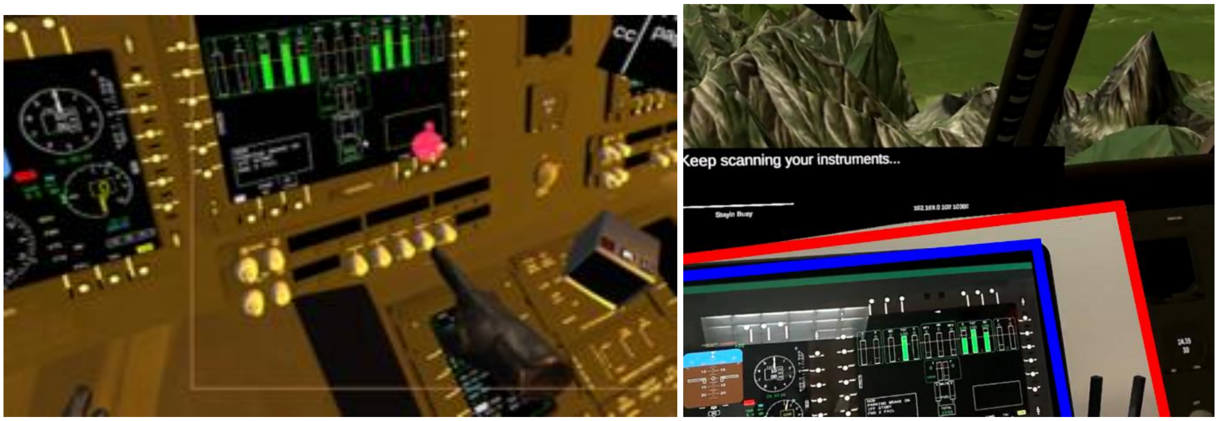


Figure 3. Physical cockpit on the left with the virtual representation on the right



Keep scanning your instruments...

Figure 4. Fully virtual system on the left and touchscreen mixed reality on the right. Cutout area of the mixed reality photo has been highlighted in red and the real-world touchscreen border highlighted in blue.

We also switched from using visual markers and Vuforia to the Magic Leap's spatial anchor system for alignment of the real-world and virtual assets. The spatial anchor system is a database of virtual anchors that are set in the real world which the Magic Leap remembers when it finds itself within the same room. Virtual objects can then be aligned to the anchors and will remain in that location each time the system is used. We then created a calibration scene which allowed pilots to place the virtual assets to locations in the real world and generate spatial anchors for their locations. This created a fast and reliable alignment system.

A commercially available 3D model of the UH-60 was downloaded and modified to match the physical cockpit used in other iterations of the simulation. The model was static so each button and dial was modified to act as a clickable object or a rotatable object. These objects were labeled in the scene with a name, for example the enter key was labeled internally as “ENT KEY”. This allowed each button to be searchable within the Unity 3D scripts and their locations used for visual cues. The back-end system could send an instruction to highlight a particular key by its internal name, and it could quickly be searched and applied to the virtual button.

We also animated the helicopter’s rotor blades to allow it to simulate takeoff. A terrain scene was loaded around the helicopter to give an environment to fly through. Volumetric clouds and particle effects were used to simulate the thunderstorm that the user would enter. The first emergency procedure would be a heading mis-compare within the instrumentation that would take five steps to complete by switching from the copilot sensor to the pilot sensor. This procedure did not result in a crash; however, it made navigation and recovery for future emergencies much harder. The next emergency, shown in Figure 5, was a severe weather event that required pilots to reroute around the storm. This event took six steps and would result in a crash if not finished in time. The final emergency procedure was a single engine failure that took two steps to correct by reducing altitude and would result in a crash if not done in time.

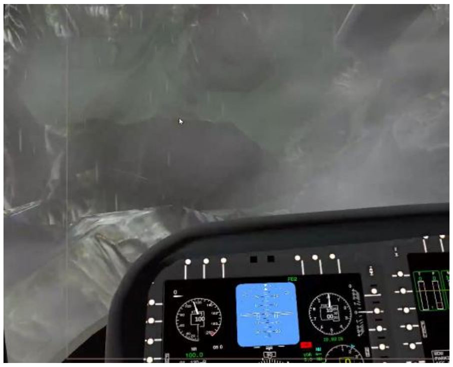
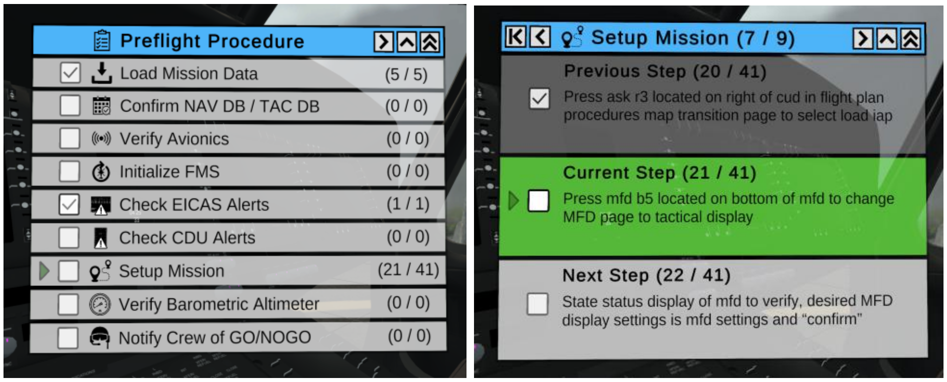


Figure 5. Example of heavy rain and lightning used to obscure vision and create distractions for the user

Unity 3D’s animation system provides a state machine system that can easily be tied to scripts and functions through visual programming. This was utilized to help maintain the state of the simulation as we moved through procedures. The state system provided realistic updates to the UH-60’s multipurpose display (MPD). These screens reflected real values for the system as the user moved through the procedures and contained animations to show heading and altitude changes of the UH-60.

In addition to the visual and audio cues telling the user what to do, a checklist, shown in Figure 6, was provided in a moveable window. This allowed the user to understand the context of their actions in relation to the procedure they were following. This checklist was meant to mimic real-world checklists that pilots use and helped in the knowledge transfer aspect of the simulation. For expert pilot flying scenarios, we hid this guidance window unless an error was detected and this window was needed for recovery. For future scenarios consideration could be given of placing the checklist is a known fixed location, such as the kneeboard or to have it appear on the hand with a simple gesture or button press.



Preflight Procedure

	Step	Progress
☑	Load Mission Data	(5 / 5)
□	Confirm NAV DB / TAC DB	(0 / 0)
□	Verify Avionics	(0 / 0)
□	Initialize FMS	(0 / 0)
☑	Check EICAS Alerts	(1 / 1)
□	Check CDU Alerts	(0 / 0)
□	Setup Mission	(21 / 41)
□	Verify Barometric Altimeter	(0 / 0)
□	Notify Crew of GO/NOGO	(0 / 0)

Setup Mission (7 / 9)

Previous Step (20 / 41)

☑ Press ask r3 located on right of cud in flight plan procedures map transition page to select load iap

Current Step (21 / 41)

□ Press mfd b5 located on bottom of mfd to change MFD page to tactical display

Next Step (22 / 41)

□ State status display of mfd to verify, desired MFD display settings is mfd settings and "confirm"

Figure 6. Examples of the procedural checklist with the full procedure on the right and individual steps on the left

Finally, a virtual assistant cube, shown in Figure 7, was added to the scene. This provided two-way audio communication between the user and the back-end system. The user could press a button on the box and speak naturally. Their audio was encoded and transmitted to the back-end system where it would be analyzed and transformed into text. It was then fed into a large language model, and the response was played back to the user via a text-to-audio system. The back-end system could also provide unsolicited advice through the same system if it noticed the user was taking an excessive amount of time to complete the step. In this scenario fNIRS data was used to determine if the user was not paying attention or if they were stressed and unable to figure out the step, which would guide the tone of the unsolicited advice message.

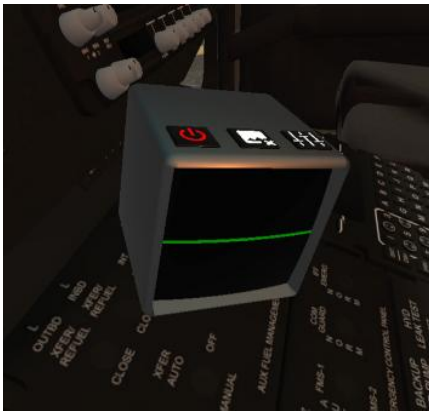


Figure 7. Virtual assistant for audio recording and playback

Operational and Look Cues

As part of our system design, we added a configuration option to the visual and operational cues used in the system.

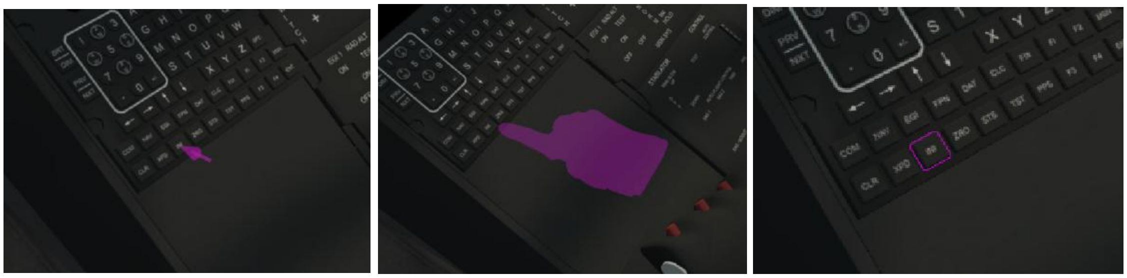


Figure 8. The Operational cues used. From left to right, pick arrow, ghost hand, and target outline

Operational cues, shown in Figure 8, guided the user to perform a certain task, such as pressing a button. These included the pick arrow (Schwerdtfeger, Reif, Gunthner, & Klinker, 2011), an arrow pointing to the object to interact with, the ghost hand, a semi-transparent hand that would point directly at the object, and the target outline, a pink box outline that would appear around the object.

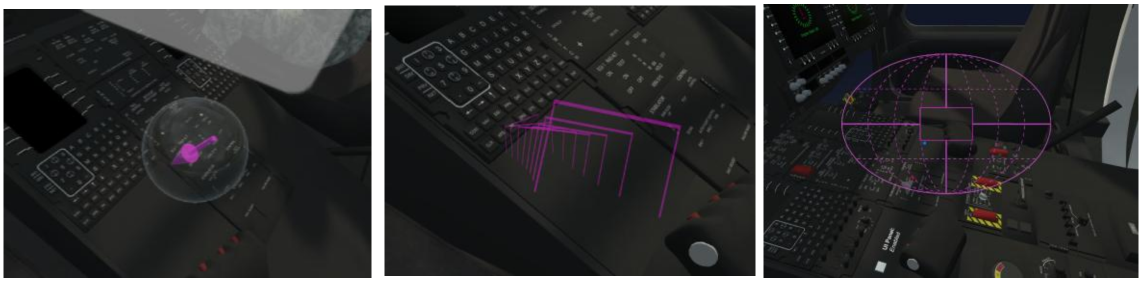


Figure 9. The Look cues used. From left to right, look arrow, attention funnel, and EyeSee360

Look cues, shown in Figure 9, would guide the pilot to look at something, for example to confirm the values they were seeing on the MPD. These cues were the look arrow, an arrow at the edge of the screen that would draw the pilot's attention, the attention funnel (Biocca, Tang, Owen, & Xiao, 2006), a 3D funnel of boxes directing the pilot's gaze, and the EyeSee360 (Gruenefeld, Ennenga, Ali, Heuten, & Boll, 2017), a projection of the 3D world and the viewer's gaze onto a map.

The configuration changes allowed us to test the scenario with multiple permutations of operational and look cues in search of an ideal combination for the UH-60 task.

Core Back-end System

The back-end architecture, shown in Figure 2, builds on a Northrop system called OCARINA. OCARINA is a robot operating system (ROS)-based framework that decomposes task-guidance intelligence into interoperable nodes that exchange messages over publish/subscribe topics. This structure allows for distributed modules across multiple hardware, as is the case with our hardware-intensive LLM module. Additionally, it allows multiple instances of the front-end to be run simultaneously, allowing us to duplicate the cockpit in both the headset and the touchscreen without modification. Lastly, it allows us to correlate multiple sensor feeds from the headset, touchscreen, and the neurophysiological sensors. We designed out a set of core modules that are required for cockpit guidance and then incorporate research modules that perform sub functionality.

The cockpit manual and its procedures are a large corpus of information that must have a structured representation to encode and understand relations. To accommodate this, domain knowledge is stored in a central Knowledge Base (KB) implemented as a tree-structured knowledge graph. Nodes encode both declarative facts (e.g. cockpit affordances, sensor quali) and procedural facts (action pre-/post-conditions) using subject-relation-object formalism. The graph supports multimodal embeddings: text, numerical types, and relational paths. Enabling a single representation for perception, reasoning, and cue generation.

To allow for subject matter experts (SMEs) to quickly encode knowledge and cockpit procedures, a lightweight, YAML-style KB language lets SMEs author or revise knowledge without writing code. Identifiers, datatypes, and hierarchical relations are auto generated, then ingested at run-time or updated dynamically as new facts are perceived. This separation of content from code makes the knowledge layer highly modifiable and version-controlled alongside procedure updates, giving it the flexibility to work in many use-case scenarios. At run-time, the KB is ingested as a graph with typed embeddings, enabling the procedural grounding AI planner and the LLM to query the same representation.

To use this knowledge stored in the KB, the world state (WS) module instantiates graph-derived objects when seen in the environment and tracks them in a single typed dictionary. These objects are updated as the world is perceived, and the dictionary can then be queried by other modules to evaluate the state of the world.

When given a goal (e.g. recover from single engine failure), the Procedural Grounding (PG) module performs goal-oriented action planning (Orkin, 2006) over the KB to create an action graph that traverses all possible ways to satisfy the pre-conditions of the goal. The PG module then traverses the graph to find the minimal action path to reach the goal given the current states of the world. The graph is continually reevaluated at millisecond resolution as the world and sensors update. This ensures that both minimal actions are presented to fix a given problem and that every action path to solve a solution is available.

In the case of the preflight checklist or during a multi-emergency scenario, multiple procedures need to be run simultaneously. Active procedures are all run in parallel and in the order that they are started. One procedure marked 'current' is presented to the pilot at a time. If the pilot performs actions on an active procedure that is not the current procedure, OCARINA automatically switches the current procedure to the new procedure on which the actions were performed. If an action is made that alters the world and is not related to an active procedure, we treat that action as an error.

To manage streamed information from the front-end, the environmental grounding (EG) module aggregates multi-sensor feeds (neurophysiology, SIM updates) and fuses processors (text-to-speech, LLM adaptation) that can be hot-swapped through configuration files. These processors are sub modules that process the environmental information in some way before publishing it. Environmental information is then queried by the world state to update its world representation.

Sub Back-end Modules

With this main pipeline, we can implement many submodules that do specific functions to our guidance. One such module we experimented with is an adapted LLM pipeline. The goal of the LLM module was to parse pilot queries and multi-modal information to provide better guidance to the pilot. On a separate server we set up a module that can take in a pilot query through front-end virtual assistant or system query when the system identifies the pilot needs additional help (multiple errors are made, time since last action threshold is exceeded), feed in the query along with encoded cockpit documents and procedures into a local LLM (Marah Abdin, 2024). This query could do one of two things: ask a question about the cockpit/procedure or ask the system to modify the guidance modality and text.

When the module is asked a question, the LLM prompt instructs the model to identify relevant areas of the cockpit manual and procedures. Then to parse that information for the pilot's question. The LLM can decide on a type of modality to send back: text/audio and/or a visual image from the manual. This information is then formatted into a parse-able format and sent back to the frontend to provide to the pilot.

When the module is asked to modify the guidance step, the LLM is provided with an updated pilot model describing recent errors, the current guidance step, physiological measures, and classified cognitive workload states of the pilot. The LLM prompt is instructed on how to use this information with expert encoded guidance adaptation examples. The module then modifies the text of the guidance step and selects a single or set of modalities to present to the pilot (textual, audible, or visual cue).

Information is given to the pilot from the LLM module in three ways: textual information is displayed to the pilot as a dialogue style box over the current guidance window; visual information is displayed on the cube; audible information is spoken to the pilot. If a pilot proceeds to the next step in a procedure, the guidance information is stopped.

EVALUATION

The system and its UH-60 use case were used in user studies, a formal evaluation, and program system demonstration.

User Studies

The UH-60 use case, front-end system, and a modified version of the back-end system were used in (Belga, et al., 2025 [Forthcoming]) to determine pilot performance to differing cues in a dense operating environment. This study was within-subjects and analyzed all permutations of operational cues with look cues along with the use of limited field of view (an issue seen within mixed reality headsets) and full field of view (VR based headsets). This created 18 conditions that 48 users with no piloting experience successfully completed. It was determined that the limited field of view of mixed reality created a statistically significant increase in completion time. Furthermore, the choice of look cue and pick cue and its efficacy depends largely on the context of the environment. In the case of a dense UH-60 environment, minimalist cues were preferred, specifically the look arrow and the pick arrow.

In another user study, (Wen, et al., 2025) integrated the UH-60 VR cockpit with AdaptiveCoPilot, a neuroadaptive back-end that qualitatively and quantitatively examined the effectiveness of the neuroadaptive guidance modules in OCARINA. Eight licensed pilots of mixed expertise executed a nine-step pre-flight checklist under three within-subjects conditions: paper checklist baseline, random cueing, and adaptive cueing. While cognitive states for working memory, perception, and attention were logged at 10 Hz. Relative to the baseline, the adaptive condition produced significantly higher odds of optimal working memory ($\beta = -0.685$, $z = -16.17$) and perception states ($\beta = -1.403$, $z = -36.48$), maintained error rates comparable to the other conditions, and showed a consistent, but nonsignificant, trend toward faster checklist completion. Post-run interviews showed that pilots valued the system's ability to scale information and switch modalities contextually, though several cautioned that overly helpful guidance risked fostering complacency during routine steps. Building on that feedback, the paper outlines four strategies for aviation guidance systems: workload-aware modality switching, redundancy-free messaging, scaffolded error-recovery cues, and experience-sensitive adaptivity. The goal of these strategies being to temper complacency while preserving trust and autonomy in next-generation adaptive flight guidance.

Formal Evaluation

Our system underwent a formal evaluation by the Massachusetts Institute of Technology (Groll & DeAngelus, 2025) and guided our final system development and lessons learned. Their evaluation looked at step recognition and error recognition. In step recognition they evaluated how well the system followed the correct steps and displayed the correct screen or response as it would be in real life (true positives), or if it missed a step completion (false negatives). Error recognition looked at its ability to detect errors and guide the pilot back to the correct steps with the correct screens or response (true negatives). Failures in this scenario were missing the error input (false positives), or incorrectly detecting the error (false negatives).

In their evaluation it was determined that our system correctly detected steps (true positives) 471 times. It had 15 false negatives during the step evaluation phase. For error detection and correction, it found the error with no issue 45 times (true negative), failed to detect the error 26 times (false positive), and incorrectly reported an error (false negative) 189 times. It was determined that the false negatives were present because of error logs generated on the back-end that were not related to the pilot's input. While counted in this study, we did not feel this degraded the system as these logs were for debugging purposes and had no effect on the pilot's interactions within the system.

In terms of correcting errors, our system was able to fully guide the pilot back to the correct procedural steps 54.2% of the time, partially guide the pilot 5.1% of the time, and failed to detect the error 40.7% of the time.

Thanks to this formal evaluation we were able to determine deficiencies between the state of the front-end system and the knowledge of the back-end system and make improvements. In the evaluation's build the back-end system acted primarily as the master system providing all instructions to the front-end system, with the front-end only sending user interactions in its communications. The evaluation showed that fast movements by the user could cause the system to fail to realize it was in an error state, and this was a reoccurring problem as users would often act quickly to try and correct their own mistakes. Given this knowledge we improved the system by increasing communication between the front-end and back-end system. Now the front-end also gave its current state of every screen along with the actions of the user. This helped the back-end system in determining the state of the user and finding the correct action path for them. As such, its ability to catch an error state and provide correct guidance back to the user.

System Demonstration

At the conclusion of the project, we performed a system demonstration to project leads at DARPA along with military stakeholders wishing to evaluate the systems. During this demonstration both novice users with no piloting experience and experienced pilots ran through both procedures. Users were given as much time as needed to complete the pre-flight procedure and had to complete nearly 250 actions to begin flying. Upon flying the user had to go through each emergency procedure: the heading mis-compare, the severe weather event, and the single engine failure. For the severe weather event they had 60 seconds to complete the procedure before crashing and for the single engine failure, they had 30 seconds to complete the procedure before crashing. The heading mis-compare did not result in a crash, only an incorrect navigation map.

Of note is that every user was able to successfully complete all tasks regardless of experience levels. For one experienced pilot they found themselves automatically reaching for controls over their head during the emergency procedure. They were reminded that they were in the co-pilot role and did not need to access those controls and that they were not present in the simulation. Still, this was noted as signs of immersion of the pilot into the training scenario. During a novice user's run through the emergency procedures, it was noted that their attention levels were overloaded through the fNIRS as they tried to locate what to do next. The back-end system used the LLM sub-module to generate a reassuring message that urged them to relax and focus, along with further guidance on what control they were looking for.

While this was an informal evaluation, it showed the efficacy of the finalized system for guiding users through complex procedures regardless of their experience levels.

CONCLUSION AND FUTURE WORK

In this paper we discussed the design and evaluation of our complex task guidance system for UH-60 pilots. During the course of the work, we went through several iterations of our front-end design finally creating one of low-cost with high immersion. The back-end system was created for easy representation of knowledge and the use of artificial intelligence to provide guidance to the pilots. Because of the separation of the systems, we were able to create a transferable back-end that has been applied to additional flight simulation use-cases not covered by this paper. During the course of this project we published six peer-reviewed papers that advanced personal intelligent systems, neuroadaptive guidance, and guidance cueing beyond just rotorcraft training.

We learned a few important lessons to follow in future collaborative guidance system work. Rigorous, distributed testing is essential for rapid prototyping. Spatial Anchoring for the pilot use case is more beneficial than a dynamic colocalization since the cockpit does not move. AR prototyping remains hardware-bound and fragile and constantly changing hardware combined with buggy integration will stall development.

While providing promising results we acknowledge there is more work to be done. Our initial evaluations did not allow pilots to run through the UH-60 procedures multiple times over an extended period. As such, we were not able to truly test knowledge transfer from the artificial expert to a novice user. In future work we would seek to gather volunteer novice pilots and have them carry out the procedures at a fixed interval (for example twice a week) for a fixed period of time (two months). We would also make adjustments to the back-end so that it could slowly ease off its guidance, only providing cues after a set period of time or when detecting an error. Over time we would evaluate how well users begin to truly understand their tasks and become less dependent on the guidance system. We would also end with a formal performance evaluation of each user with no guidance to test complete knowledge transfer.

The system developed was important for the research of multiple papers and its evaluation, both formally and informally, shows the efficacy of the approach in task guidance. Modern technology, such as mixed reality technology and artificial intelligence, can be combined for the complex problem of task guidance at a low cost. Furthermore, we believe that our future work will show its capabilities at complete knowledge transfer from an artificial expert agent to a novice user. We also showed the capability of the system as a passive error monitor for experienced pilots, offering advice only when needed. While in its initial stage, our prototype shows the possibilities of creating a low-cost, transferrable, training and error monitoring system.

ACKNOWLEDGEMENTS

Approved for Public Release; NG25-1185 Northrop Grumman Systems Corporation. This work was supported by the DARPA PTG program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

REFERENCES

Barter, S., Clayton, J., & Clark, G. (1993). Aspects of fatigue affecting the design and maintenance of modern military aircraft. *International journal of fatigue, 15*(4), 325-332. doi:10.1016/0142-1123 (93)90382-Z 2

Belga, J., Ghamandi, R., Chawla, N., McKendrick, R., McMahan, R., & Laviola, J. J. (2025 [Forthcoming]). An Investigation of Look Cues and Pick Cues for Guidance in Dense Operating Environments. *The Human Factors and Ergonomics Society*.

Biocca, F., Tang, A., Owen, C., & Xiao, F. (2006). Attention funnel: omnidirectional 3D cursor for mobile augmented reality platforms. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 1115-1122.

Cao, S., Kearns, S. K., Niechwiej-Szwedo, E., & Irving, E. (2024). Computational cognitive modeling of pilot performance in pre-flight and take-off procedures. *Journal of Aviation/Aerospace Education & Research, 33*(4), 1-21. doi:https://doi.org/10.58940/2329-258X.2026

Castelo, S., Rulff, J., Solunke, P., McGowan, E., Wu, G., Roman, I., . . . Silva, C. (2024). Hubar: A visual analytics tool to explore human behavior based on fnirs in ar guidance systems. *IEEE Transactions on Visualization and Computer Graphics*, 119-129.

Degani, A., & Wiener, E. L. (1993). Cockpit checklists: Concepts, design, and use. *Human Factors*, (pp. 345-359).

Garvey, T. D., Gervasio, M. T., Lee, T. J., Myers, K. L., Angiolillo, C., Gaston, M. E., . . . Kolojejchick, J. (2009). Learning by Demonstration to Support Military Planning and Decision Making. *IAAI* (pp. 85-92). AAAI.

Groll, M. F., & DeAngelus, M. A. (2025). *DARPA PTG Evaluation of NGC Ocarina.* MIT Lincoln Laboratory Internal Report. Available from MIT Lincoln Laboratory upon request.

Gruenefeld, U., Ennenga, D., Ali, A. E., Heuten, W., & Boll, S. (2017). Eyesee360: Designing a visualization technique for out-of-view objects in head-mounted augmented reality. *Proceedings of the 5th symposium on spatial user interaction*, 109-118.

Hughes, C. E., Stapleton, C. B., Hughes, D. E., & Smith, E. M. (2005). Mixed reality in education, entertainment, and training. *IEEE computer graphics and applications. 25*, pp. 24-30. IEEE.

Kimball, D. R., & Holyoak, K. J. (2000). Transfer and expertise. *The Oxford handbook of memory*, 109-122.

King, G., Carmody, K., & Deaton, J. (2023). The Influence of New Realities: How Virtual, Augmented, and Mixed Reality Advance Training Methods in Aviation. In D. A. Vincenzi, M. Mouloua, H. Peter, J. A. Pharmer, & J. C. Ferraro, *Human Factors in Simulation and Training* (pp. 317-330). CRC Press.

Knerr, B. W. (2006). Current issues in the use of virtual simulations for dismounted soldier training. *Virtual Media for Military Applications*, 1-12.

Kryskow, M. A., Beidleman, B. A., Fulco, C. S., & Muza, S. R. (2013). Performance during simple and complex military psychomotor tasks at various altitudes. *Aviation, space, and environmental medicine, 84*(11), 1147-1152.

LaViola, J., Williamson, B., Brooks, C., Veazanchin, S., Sottilare, R., & Garrity, P. (2015). Using augmented reality to tutor military tasks in the wild. *IITSEC*, (pp. 1-10).

Macchiarella, D., Yu, J., Liu, D., & Vincenzi, D. A. (2023). Augmented reality as a means of job task training in aviation. In D. A. Vincenzi, M. Mouloua, H. Peter, J. A. Pharmer, & J. C. Ferraro, *Human Factors in Simulation and Training* (pp. 65-86). CRC Press.

Marah Abdin, J. A. (2024). Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone.

McGowan, E., Rulff, J., Castelo, S., Wu, G., Chen, S., Lopez, R., . . . Silva, C. (2025). Design and Implementation of the Transparent, Interpretable, and Multimodal (TIM) AR Personal Assistant. *IEEE Computer Graphics and Applications*, 28-42.

McKendrick, R. a. (2019). Theories and Methods for Labeling Cognitive Workload: Classification and Transfer Learning. *Frontiers in Human Neuroscience*, 295.

Nadri, C., Liu, Y., Zahabi, M., Kaber, D., Ruiz, J., Middleton, M., & McKendrick, R. (2024). Analysis of Pre-Flight and Monitoring Tasks Using Cognitive Performance Modeling. *Human Factors in Design, Engineering, and Computing.* AHFE Open Acces.

Ockerman, J. J., & Pritchett, A. R. (1998). Preliminary investigation of wearable computers for task guidance in aircraft inspection. *Second International Symposium on Wearable Computers* (pp. 33-40). IEEE.

Orkin, J. (2006). Three States and a Plan: The A.I. of F.E.A.R. *Game developers conference.*

Peintner, B., Dinger, J., Rodriguez, A. C., & Myers, K. L. (2009). Task Assistant: Personalized Task Management for Military Environments. *IAAI* (pp. 128-134). AAAI.

Schaffernak, H., Moesl, B., Vorraber, W., Holy, M., Herzog, E.-M., Novak, R., & Koglbauer, I. V. (2022). Novel mixed reality use cases for pilot training. *Education Sciences, 12*(5), 345-363.

Schlichting, M. R., Rasmussen, V., Alazzeh, H., Liu, H., Jafari, K., Hardy, A. F., . . . Kochenderfer, M. J. (2025). LeRAAT: LLM-Enabled Real-Time Aviation Advisory Tool. *arXiv:2503.16477*.

Schutte, P. C., & Trujillo, A. C. (1996). Flight crew task management in non-normal situations. *HFES. 40*, pp. 244-248. SAGE.

Schwerdtfeger, B., Reif, R., Gunthner, W. A., & Klinker, G. (2011). Pick-by-vision: there is something to pick at the end of the augmented tunnel. *Virtual Reality, 15*, 213-223.

Shervedani, A. M., Walter, M. R., & Zefran, M. (2025). From Vague Instructions to Task Plans: A Feedback-Driven HRC Task Planning Framework based on LLMs. *arXiv*, 1-8.

Sie, L. a. (2009). Passion and expertise knowledge transfer. *Journal of Knowledge Management, 13*(4), 175-186.

Socha, V., Socha, L., Hanakova, L., Valenta, V., Kusmirek, S., & Lalis, A. (2020). Pilots' performance and workload assessment: Transition from analogue to glass-cockpit. *Applied Sciences, 10*(15).

Sottilare, R. (2013). Adaptive Intelligent Tutoring System (ITS) Research in Support of the Army Learning Model. *US Army Research Laboratory*.

Spector, J. M. (2008). Expertise and dynamic tasks. In H. Qudrat-Ullah, J. M. Spector, & P. Davidsen, *Complex decision making: Theory and practice* (pp. 25-37). Springer.

Uhrmann, J., Strenzke, R., & Schulte, A. (2010). Task-based guidance of multiple detached unmanned sensor platforms in military helicopter operations. *COGIS*.

Wang, L., Chou, J., Tien, A., Zhou, X., & Baumgartner, D. (2024). AviationGPT: A large language model for the aviation domain. *AIAA Aviation Forum*, (pp. 1-14).

Wen, S., Middleton, M., Ping, S., Chawla, N. N., Wu, G., Feest, B. S., . . . McKendrick, R. (2025). AdaptiveCoPilot: Design and Testing of a NeuroAdaptive LLM Cockpit Guidance System in both Novice and Expert Pilots. *2025 IEEE Conference Virtual Reality and 3D User Interfaces* (pp. 656-666). IEEE.

Wu, G., Qian, J., Castelo Quispe, S., Chen, S., Rulff, J., & Silva, C. (2024). ARTiST: Automated text simplification for task guidance in augmented reality. *CHI Conference on Human Factors in Computing Systems*, (pp. 1-24).