Available online at http://docs.lib.purdue.edu/jate

JATE

Journal of Aviation Technology and Engineering 8:1 (2018) 31–50

Operating Different Displays in Military Fast Jets Using Eye Gaze Tracker

Jeevitha Shree DV, L. R. D. Murthy, Kamalpreet Singh Saluja, and Pradipta Biswas

Indian Institute of Science

Abstract

This paper investigated the use of an eye-gaze-controlled interface in a military aviation environment. We set up a flight simulator and used the gaze-controlled interface in three different configurations of displays (head down, head up, and head mounted) for military fast jets. Our studies found that the gaze-controlled interface statistically significantly increased the speed of interaction for secondary mission control tasks compared to touchscreen- and joystick-based target designation system. Finally, we tested a gaze-controlled system inside an aircraft both on the ground and in different phases of flight with military pilots. Results showed that they could undertake representative pointing and selection tasks in less than two seconds, on average.

Keywords: gaze-controlled interface, flight simulator, adaptive MFD, eye-gaze tracker, aviation UI

Introduction

Advancement of sensor and communication technologies has made aviation easier and safer at a cost of generating a huge amount of information from aircraft. Although a good amount of information is used for offline processing on the ground or automated processing by mission computers on board, like controlling the autopilot system, pilots need to manually perceive and process a plethora of information for decision making for both flying and mission control tasks (Hierl, Neujahr, & Sandl, 2012). Information processing is more difficult in military fast jets (fighter aircraft used for air superiority or multirole missions) than passenger aircraft as pilots need to undertake secondary mission tasks in addition to the primary flying task. Secondary mission control tasks may include reconnaissance, protecting or tracking airborne assets, and weapon delivery, all of which require careful perception and analysis of information outside the aircraft as well as information displayed in the cockpit. Efficiently displaying information inside the limited space of the cockpit is a challenging design task. Existing military aircraft use three types of visual display: head-down display (HDD), head-up display (HUD), and head-mounted display (HMD).

HDDs are configured to display information as multifunction displays (MFDs). MFDs are used for showing information ranging from primary flight data to details of airborne objects in a configurable way. Each is rectangular in shape, consists of a set of soft buttons on the periphery, and a graphical user interface (GUI) in the middle. MFDs can present data in multiple pages, rather than always being present at once like analog displays and so consumes less space in a cockpit compared to analog displays.

HUDs are placed in the line of sight of the pilot, collimated at infinity to reduce parallax error (Wood & Howells, 2001). HUDs show primary flight information like speed, altitude, azimuth level, and airbourne targets. HUDs do not directly take

http://dx.doi.org/10.7771/2159-6670.1184

input from users like MFDs but update information appropriately as the aircraft is operated.

HDDs and HUDs are fixed in place in the cockpit while HMDs are projected from the helmet and pilots can get information even if they are not looking straight ahead. HMDs work like an augmented reality system. Pilots do not loose situational awareness when using HMDs; they can see through the display while it displays extra information on primary flight parameters and airborne objects in a see-through display.

In terms of taking input from users, HDDs are operated using physical touch and target designation system (TDS), which is a joystick attached to the throttle (Eurofighter, 2017). Recently, voice commands (Hierl et al., 2012) were also used by uttering the caption on buttons. HMDs are also operated using the TDS.

Existing military aircraft have multiple HDDs and physically touching an MFD in an HDD with gloves may often become difficult for small targets (Hierl et al., 2012). The TDS can be improved by reducing the pointing time (the time needed to bring a cursor on a screen element). Target prediction algorithms, which require *a priori* location of targets on screen (Dixon, Fogarty, & Wobbrock, 2012), cannot be used for MFDs as targets may appear anywhere on the GUI part (here the GUI part indicates the portion of the MFD between the peripheral Bessel keys). There are also limited interaction modalities available as pilots find it difficult to take their hands away from throttle and flight stick in high-G manoeuvers (Hierl et al., 2012). Direct voice input has already been tested in military aviation, although voice recognition systems are challenging for non-native English speakers and for countries having multiple language speakers (Biswas & Langdon, 2015; Hierl et al., 2012).

This paper proposes the use of an eye gaze control system to improve existing TDSs. Operators have to look at various displays to process information such as investigating positions of airborne targets with an MFD or an HMD or changing pages of an MFD. We propose leveraging this eye gaze as a direct way of controlling pointer movements on an MFD and an HMD as opposed to touchscreen or HOTAS (hands on throttle and stick) thumbstick. Eye gaze trackers as direct controllers of pointer movement have not yet been widely tested in the military aviation environment. De Reus, Zon, and Ouwerkerk (2012) reported qualitative data on using a gaze-controlled interface in a helmet but did not specify the accuracy of the gaze tracker or any quantitative data on processing speed of information. According to TopGunMilitary (2013), the gaze control system of the M-TADS system in Apache AH-64 attack helicopters allows the pilot to aim his or her weapon systems by merely looking at the desired target.

We tested an eye-gaze-controlled interface with a flight simulator and proposed a couple of adaptation algorithms for gaze-controlled displays. Our algorithm significantly reduced pointing and selection times for large peripheral buttons as well as improved flying performance for the gaze-controlled system compared to touchscreen for representative flying and pointing and selection tasks.

We also investigated using MFDs in HUD instead of HDD and designed a frugal eye-gaze-controlled HUD. Our proposed algorithms significantly reduced pointing time and improved flying performance in terms of deviation from flight envelope compared to an existing TDS in representative dual task studies described later.

Finally, we used a wearable gaze tracker and simulated a case study of operating an HMD using gaze tracking. We noted a one-third decrease in pointing and selection times for a gaze-controlled system compared to a traditional TDS. The main contributions of the paper are

1. Integrating and exploring an eye-gaze-controlled interface for a flight simulator.
2. Proposing new algorithms to improve pointing and selection times in a gaze-controlled interface.
3. Designing and evaluating gaze-controlled HUD and simulated HMD displays.

The paper is organized as follows. Section 2 presents related literature on gaze-controlled interface and pointing facilitation systems. Sections 3 describes our flight simulator setup followed by a couple of pilot studies involving a military aircraft and high-end third party simulator in Section 4. Sections 5 to 8 present three user studies using our flight simulator setup and a study conducted in an aircraft both on the ground and in the air. Section 9 presents a general discussion followed by concluding remarks in Section 10.

Related Work

Eye tracking is the process of measuring either the point of gaze (where one is looking) or the motion of an eye relative to the head. An eye tracker is a device for measuring eye positions and eye movement. Although research on eye tracking dates back to the early 20th century (Huey, 1908), controlling a display with gaze tracking is a relatively new concept. Eye-gaze-controlled interfaces were mainly explored as assistive technology for people with severe motor impairment. However, using eye tracking in situational impairment poses new challenges in terms of speed and accuracy. A detailed survey on gaze-controlled interfaces can be found elsewhere (Biswas & Langdon, 2015; Biswas, Prabhakar, Rajesh, Pandit, & Halder, 2017; Zhai, Morimoto, & Ihde, 1999). Existing research on comparing eye-gaze-controlled interfaces with other modalities is mainly limited to desktop computing and, except a few cases involving novice users (Biswas & Langdon, 2014), generally traditional input devices like mouse or touchscreen worked better than gaze-controlled systems (Vertegaal, 2008; Ware & Mikaelian, 1987). A few recent studies on automotive user interfaces found gaze-controlled interfaces worked worse than (Poitschke, Laquai, Stamboliev, & Rigoll, 2011) or similar to (Biswas et al., 2017) touchscreen in terms of pointing and selection times although with higher rate of errors.

Researchers have already explored various target prediction or intent recognition techniques for reducing pointing times. Most of these techniques continuously record velocity, acceleration, and bearing of cursor movement, fit different models, and use that model to predict either the cursor movement or the target itself (Murata, 1998; Pasqual & Wobbrock, 2014; Ruiz & Lank, 2010). However, eye gaze movements are not as smooth as mouse, finger, or hand movements, but rather follow a "spotlight" metaphor and so far no next point or target prediction models have been tested for gaze-controlled interfaces. A backpropagation neural network model was used (Biswas & Langdon, 2015), but only to differentiate between ballistic and homing movements. Farrell and Zhai (2005) noted that "humans use their eyes naturally as perceptive, not manipulative, body parts. Eye movement is often outside conscious thought, and it can be stressful to carefully guide eye movement as required to accurately use these target selection systems." Gaze tracking is used for zooming or accentuating part of a display to help pointing by another pointing modality (Jacob, Hurwitz, & Kamhi, 2013, Pfeuffer & Gellersen, 2016; Zhai et al. 1999).

In the military aviation domain, the United States Air Force (Furness, 1986) idealized a "Super Cockpit," a generic crew station that would conform to operators' natural perceptual, cognitive, and motor capabilities. Technologies were envisaged that would provide the means to create virtual worlds with visual and auditory fidelity, and perceive operators' interactions using cognitively simpler, "biocybernetics" control inputs. In recent times, a similar concept was considered by BAE Systems (2018) as a sixth-generation cockpit that will use the latest AR/VR technologies with head tracking and gesture recognition features. None of these idealized models considered the gaze-controlled interface. To date, aircraft use hardware switches, joysticks, auditory feedback, and small-screen displays (primary flight display or MFD) as main input and output modalities. Recently, touchscreen displays and head trackers have also been explored in fighter aircraft. Thomas, Biswas, and Langdon (2015) listed all input and output modalities used in a few modern fighter aircraft. Grandt, Pfendler, and Mooshage (2013) compared trackball, touchscreen, speech input, and mouse for an anti-warfare system; however, the experimental set up was similar to a desktop computing environment with pointing and selection as the only task to be undertaken. Thomas (2018) reported results from ISO 9241 task involving thumbstick, trackball, touchpad, and touchscreen and found trackpad to be most accurate and argued for supporting new modalities for HOTAS cursor control device. Biswas and Langdon (2015) reported a significant reduction in reaction time for a gaze-controlled interface compared to HOTAS TDS for operating a simulated MFD. However, that study did not involve any primary flying task and was configured in a desktop computing environment.

Gaze-controlled interface has also been explored for virtual and augmented reality systems. Commercial eye gaze trackers have been integrated with virtual reality systems (Ergoneers, 2018; Tobii, 2018b). For augmented reality-based systems, researchers mainly explored detection of eye gaze, but the resulting gaze tracking system is not as accurate as existing commercial systems (Handa & Ebisawa, 2008; Plopski et al., 2015). For example, Handa and Ebisawa (2008) measured accuracy only with fixed head position and reported results only in terms of nine points on screen, while commercial wearable gaze trackers (Tobii, 2018a) have accuracy at 0.5° of visual angle. Existing HMDs like the BAES Striker® II system (Striker, 2018) works based on head tracking and augmented reality. It augments pilot vision with extra information in see-through display: the head trackers are used to put the display in a pilot's visual field. HMDs display weaponry information and airborne objects and need actuation from a HOTAS stick to select the appropriate weapon to engage. This paper explored if one can use a commercial wearable gaze tracker as a cursor control device, thus eliminating HOTAS-based actuation for operating HMDs.

Flight Simulator Setup

We designed a flight simulator to conduct studies in dual task settings involving gaze-controlled interfaces. Using our setup, participants undertook standard flying tasks in parallel with representative pointing and selection tasks. This setup allowed us to measure not only pointing and selection times, but also the total response time, consisting of the time required to switch from primary flying to secondary mission control tasks. We set up the flight simulator with both HDD and an interactive HUD. Existing high-end simulators are meant for pilot training purposes and do not allow configuring MFDs to explore different adaptation strategies due to software integration, time duration, and security reasons. They also do not have any interactive HUDs. To our knowledge, no HUDs from either the automotive or aviation domain take input from users; they are only used to display primary flight information. We explored the possibility of displaying MFD at the same visual line of sight of pilots' normal seating position, so that they never need to look down for any information. Presently, HMDs are designed to interact with a display without losing situational awareness (Eurofighter, 2017); our results on HUDs can also be extended for HMDs.

The experimental setup consisted of primary and secondary displays. The flight simulator was projected on a screen. The secondary task (described below) was displayed either on an LCD screen or on a projected display configured as an HUD. The projected display (Figure 1) consisted of a semi-transparent 8 mm polyethylene sheet and a stand to hold it in front of the pilot with minimal occlusion of the situation in front. A support structure was developed in order to keep the thin plastic screen upright along with a support for the eye gaze tracker. The projected display system was configured so that its distance from the projector and angle of the eye tracker were adjustable for individual pilots.

We did not find any commercial product involving an eye-gaze-controlled head-mounted augmented reality system. There are research products with optical see-through displays with eye tracking option (Handa & Ebisawa, 2008; Lewis et al., 2013; Plopski et al., 2015) and commercial products with eye-gaze-controlled virtual reality system and helmet-mounted eye-gaze tracker (Ergoneers, 2018; Tobii, 2018b). We set up a display on the side of the pilot's position (Figure 2c) to simulate an HMD and used a wearable gaze tracker to control the display (Figure 2c). The display can be set up at any position with respect to the pilot's position. In this paper, we explored whether we can:

- Use a wearable gaze tracker as a cursor control device.
- Use it on a sidewise display or in a display which is not directly in front of the operator.
- Use it in a flight simulator.
- Control a sidewise display while undertaking a primary flying task.

At this point, we did not try to develop a new HMD or use an existing HMD. However, the wearable gaze tracker with our cursor control algorithm can be used with side-mounted HMDs or any HMD that does not look like a spectacle. We describe details of the algorithm to control a cursor using the wearable gaze tracker in a later subsection.

The LCD screen, or the HUD, or the simulated HMD were used, but not all together (Figure 2). The secondary projection display is positioned in the line of sight with the primary projection display. Table 1 furnishes the dimensions and resolutions of all three displays.

Third-party flight simulator YSFlight with data logging feature was configured for this study as an F/18 E/F Super Hornet aircraft. The flight simulator was configured with a Thrustmaster Warthog HOTAS (HOTAS, 2017). Both flight simulator and secondary pointing tasks were run on an Intel® Pentium® CPU G3220@3GHz computer running Windows 7 operating system with 4GB RAM and NVidia GeForce 210 graphics card. A Tobii X3 eye gaze tracker was used for eye gaze tracking. Senior pilots with ranks ranging from air marshall to wing commander operating fast jets for the national air force were invited to operate the flight simulator and their comments and suggestions were implemented in terms of flight stick controls and secondary task design. The size of the MFD was also set up based on the literature (Eurofighter, 2017) and feedback from Hindustan Aeronautics Limited (HAL). During trials, participants don a helmet and pair of gloves supplied by HAL (Figure 3). The large peripheral buttons were 1.7 cm square, while small buttons were 0.5 cm diamonds.

Flying Task: A map was configured with a straight line drawn in the middle. Participants were instructed to fly between 1000 and 2000 feet along the straight line. The secondary task was initiated after the flight reached the designated flight envelope of 1000 and 2000 feet.

Secondary Task: In all our subsequent studies, users undertook pointing and selection tasks in the secondary display while flying. Pointing and selection tasks involved selecting a random button on the secondary display (Figure 4). The secondary task was initiated with an auditory cue.

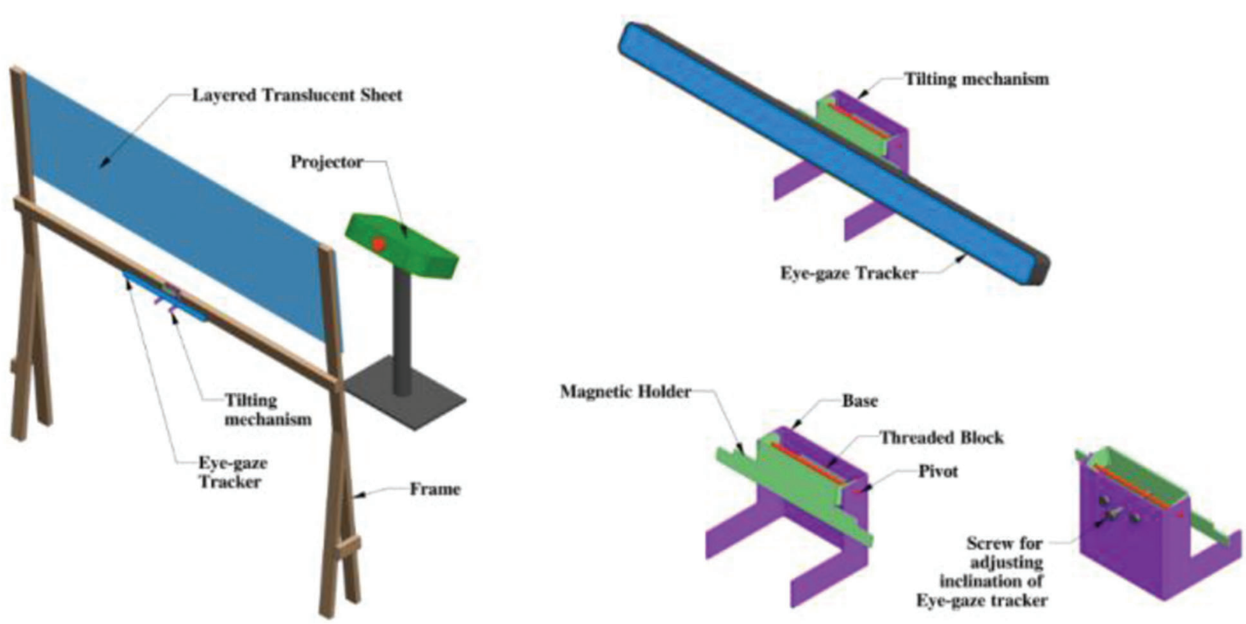
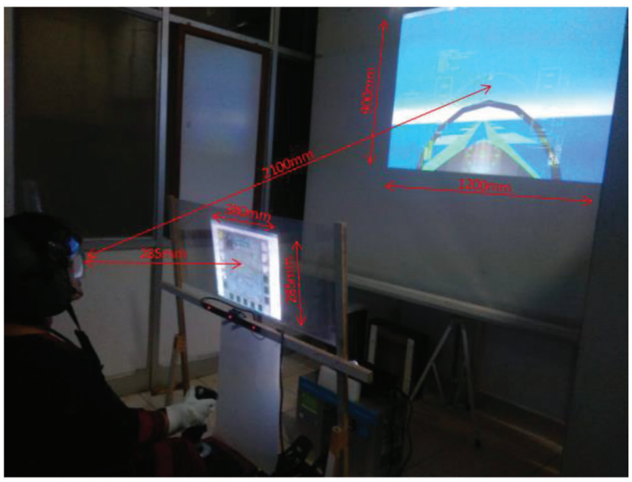


Figure 1. Projected display used as HUD.

a. Head Down Display



b. Head Up Display

c. Simulated Head Mounted Display

Figure 2. Flight simulator setup.

On hearing a beep sound, participants were instructed to undertake one button selection task on the secondary display and then switch their attention back to flying until they heard the next auditory cue (beep sound).

- For HDD and HUD, we rendered an MFD on the secondary display. The MFD consisted of 17 peripheral buttons and 10 small buttons at the central GUI part. Users needed to select one of these 27 buttons.
- For HMD, we rendered only five small buttons in a sky-colored background and users needed to select one of these five buttons.

Target Designation System: A joystick on the throttle of the HOTAS was configured as the TDS. For gaze-controlled interface, the TDS was only used for selection, not for moving the pointer on the MFD.

Eye-Gaze-Controlled HDD

In the following subsections, we describe a set of intelligent algorithms used to control a GUI with eye gaze tracker and facilitate pointing and selection tasks for various aviation displays.

Controlling GUI through screen-mounted eye gaze tracker: After developing the flight simulator, we developed the following algorithm for controlling an on-screen cursor using a screen-mounted eye gaze tracker. Retinal ganglion cells only respond to change in stimuli, and even if eye gaze is focused at a point, microsaccadic eye gaze movements always change the gaze location around the point of interest. Due to these microsaccadic eye gaze movements, the raw output from any eye gaze tracker cannot be used to control an on-screen pointer and it requires filtering to stabilize the cursor. Our gaze tracking system records the eye gaze positions continuously (see Figure 5, point A) and takes the median of the pixel locations in every 250 ms (minimum duration of a saccade) to estimate the region of interest or saccadic focus points (point B in figure 5). The median was less susceptible to outliers than the arithmetic mean in the case where the eye gaze tracker briefly lost signal. We simulate the eye movement using a Bezier curve that smooths the cursor movement between two focus points as we wanted to make the cursor movement look like an existing TDS. It then pushes the focus points into a stack and the Bezier curve (Salomon, 2005) algorithm interpolates points between the two focus points (see Figure 5, point B). The pointer is drawn at each interpolated point in every 16 ms to visualize a smooth on-screen movement (see Figure 5, point C). For adaptive conditions, it activates or enlarges the on-screen target nearest to the present gaze location (see Figure 5, point C). Using this algorithm, we can produce a smooth cursor trajectory through the otherwise jittery eye gaze movements.

Adaptation for Large Buttons of an MFD: In user studies involving the gaze-controlled interface, we noted that as users stare at the middle of the target, due to the inaccuracy of the tracker or their head movement, the neighboring button was occasionally selected. The probability of wrong selection increases if the buttons are

Table 1
Specifications of displays.

	Primary display	**Secondary display for HUD**	**Secondary display for HDD**	**Secondary display for HMD**
Projector/monitor model	Dell 1220	Acer X1185G	Waveshare 10.1″ LCD	Acer X1185G
Resolution (pixel)	1024 × 768	1024 × 768	1280 × 800	1024 × 768
Size of projection/monitor (mm)	1200 × 900	380 × 295	250 × 167	380 × 295
Distance from eye (mm)	2100	285	280	813

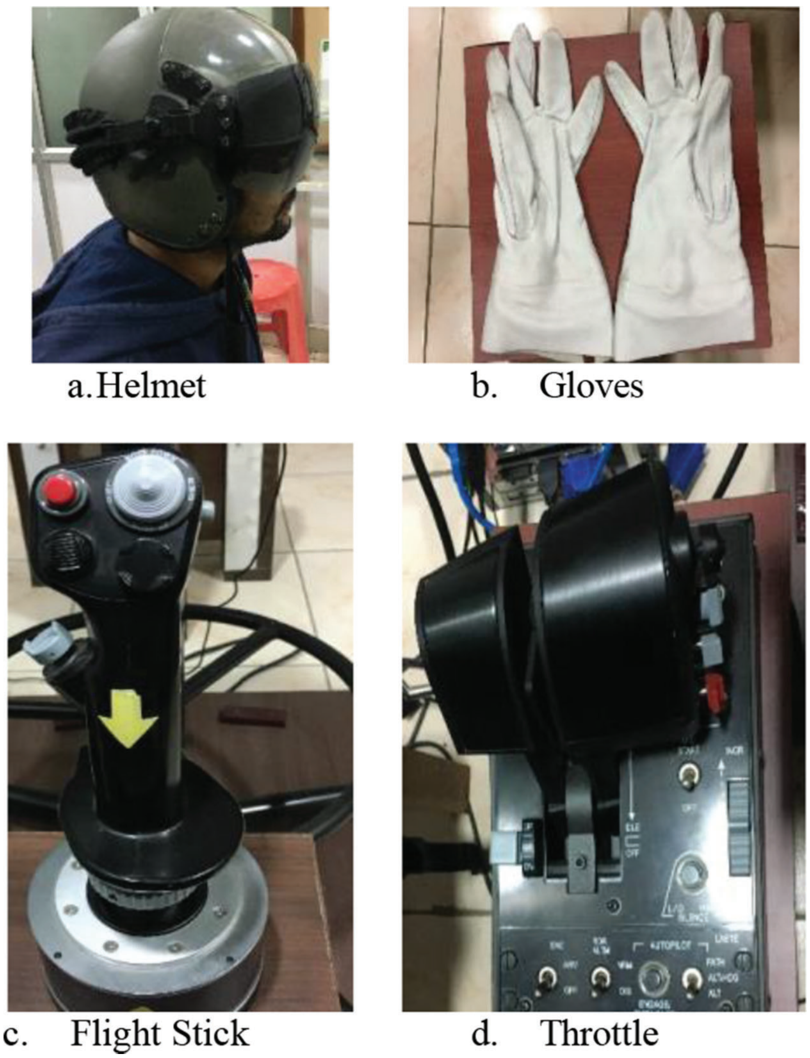


Figure 3. Instruments used in flight simulator.

closely spaced in the interface. Hence, the probability of wrong selection will be reduced if we can increase the inter-button spacing. However, we cannot change the design of an interface like an existing MFD just to make an interaction technique work better.

We have explored the option of introducing hotspots inside each button to facilitate eye gaze tracking interaction. If we can introduce a hotspot on each button and keep them well separated, we can instruct users such that the first saccade on a button would launch on these hotspots. We hypothesize that keeping these hotspots well separated will reduce pointing times and chances of wrong selection.

To find the best position of hotspots, we represented an interface as a graph where each node corresponds to a target button (clickable objects) and neighboring buttons are connected by an edge. We assumed each button has a hotspot on it, which is initially located at the center of the button. The weight of each edge is equal to the Euclidian distance between the hotspots of two neighboring buttons. We explored two different algorithms to increase the

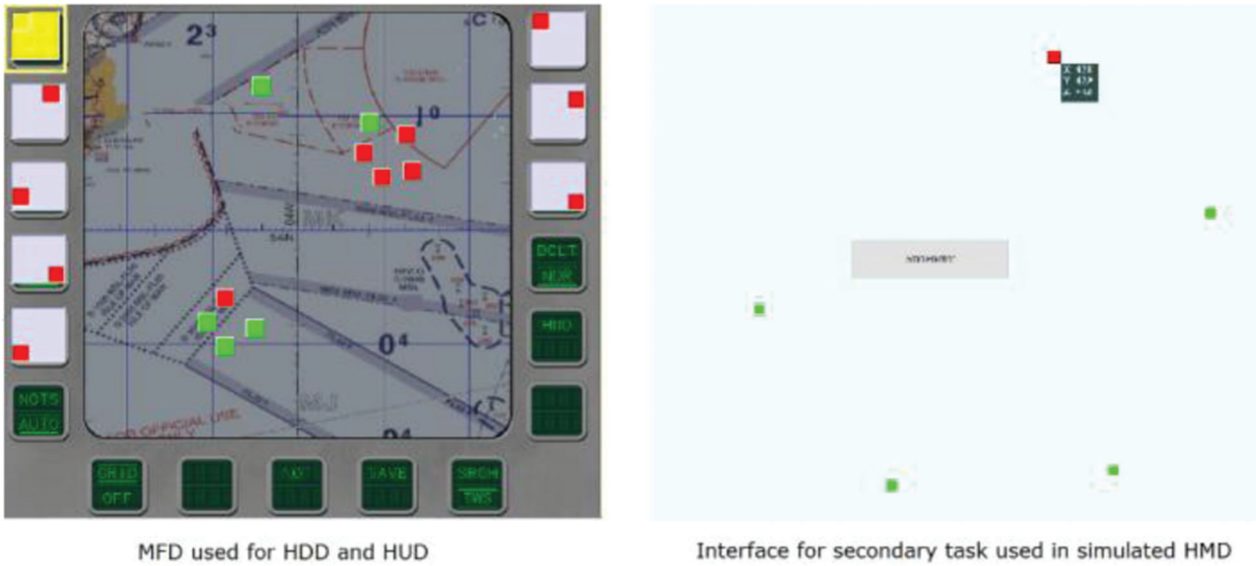


Figure 4. Screenshot of secondary tasks.

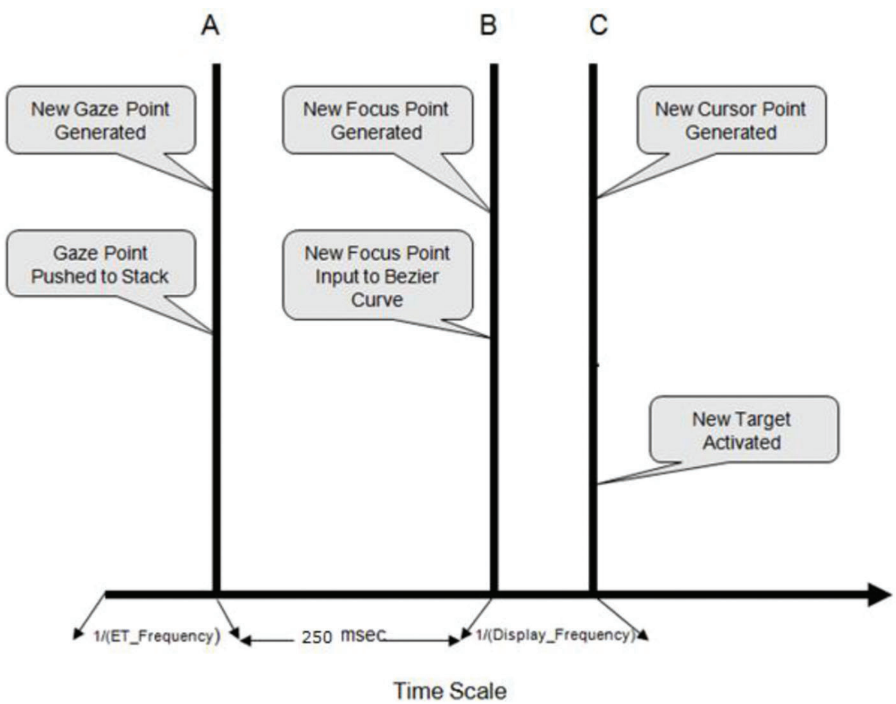


Figure 5. Cursor control through eye gaze.

distances between hotspots. We defined the following cost function and tried to minimize it:

$$Cost\,function = \sum\nolimits_{\forall d_{ij}} \frac{1}{d_{ij}^2}, \quad (1)$$

where d_{ij} is the distance of the hotspots between buttons i and j and is equal to the weight of the edge between nodes i and j.

We have modeled the problem of finding optimum locations of hotspots as a state space search problem. Each state corresponds to a particular organization of hotspots. A state transition occurs when any hotspot changes its position. If we consider each button has k possible positions and if an interface has n buttons, then an exhaustive search algorithm is needed to evaluate k^n states. Even for a moderately complex interface, an exhaustive search algorithm will be computationally intensive or almost impossible. Hence, we used the following two algorithms.

Greedy Algorithm: This algorithm picks the edge with minimum weight, which means the two most closely spaced buttons. It checks the degrees of the two nodes of the minimum-weight edge and updates the hotspot of the node with higher degree. The algorithm calculates the centroid of the hotspots of neighboring nodes of the selected node and the new hotspot is calculated as the nearest point on the selected button (or node) to the centroid. While selecting the next node for updating hotspot, the algorithm checks whether the node was visited earlier and, if so, it selects a different node. The algorithm is greedy in the sense that it only updates the hotspot if the overall value of the cost function is reduced from the previous value.

Simulated Annealing: This algorithm randomly selects a node and also randomly selects a point on the node as its new hotspot. If the new hotspot reduces the value of the cost function, then it is selected and updated. However, even if the new hotspot increases the value of the cost function, it may still be selected based on the following condition:

$$e^{\frac{(oldCostFn - newCostFn)}{T}} > \text{a random number between 0 and 1} \quad (1)$$

In the above equation, the value of T runs from 5000 to 1 and is reduced by 1 in each iteration.

We have tested the algorithm on a sample MFD and tested the algorithms multiple times with different initial positions of the hotspots. It may be noted that both algorithms reduced the cost function and increased the overall weight of edges (Figure 6). The simulated annealing algorithm reduced the cost function further than the greedy algorithm. The greedy algorithm was stuck in cycle and a local optimum after visiting all nodes a couple of times. The simulated annealing algorithm can overcome a local optimum due to randomly choosing node and hotspots, although we also could not conclude whether it reached the global optimum. It may be noted that the weights of edges in the final state of simulated annealing were significantly lower (t (0,13) = 3.2, $p < 0.01$) than those in the initial state in a paired t-test. Figure 7 shows an example of a set of hotspots on a sample MFD. The blue dots on the buttons are obtained through the simulated annealing algorithm discussed above.

Adaptation for Small Buttons: The previous strategy of introducing hotspots did not work for small buttons used to indicate airborne objects, as the buttons are too small to hold a hotspot. Additionally, these buttons may appear at a higher density than the accuracy of an eye gaze tracker.

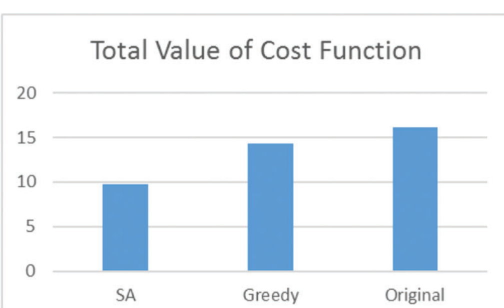


Figure 6. Comparing state space search algorithms.

We developed an adaptive zooming technique to increase inter-button spacing of densely packed on-screen elements. In desktop computing, zooming a part of a display for reducing pointing times has already been well explored (McGuffin & Balakrishnan, 2005) and for gaze-controlled interfaces, too (Bates, 1999). In our experimental setup, we used 175% zooming, as it resulted in an inter-button spacing more than 0.4° of visual angle (Figure 8), which was the accuracy of our eye gaze tracker. In actual implementation, whenever a new target appears, or an existing one changes position, we calculate the pairwise distance among all pairs of targets and decide the zooming level based on the minimum distance. We implemented the following two techniques for selecting small targets in the MFD.

Adaptation 1: In this technique, if users click near a small button, a 250 × 250 pixel area around the button enlarges in size. In this experimental setup, we hard-code the zooming percentage and zooming area, although both can be configured dynamically. In the zoomed-in area, a nearest neighborhood predictor was implemented: users could select a target by clicking near the target.

Adaptation 2: This technique was similar to the previous technique with one more feature in the implementation of the nearest neighborhood predictor. In this technique, users only need to stare (or look) near the target item and, upon dwelling for one second, the target is automatically selected. Additionally, users could also select a target by clicking near the target, as in the previous strategy.

In all adaptation strategies, the target button was indicated by drawing a yellow ring around it. The yellow ring was made distinctive so that users did not need to undertake a serial visual search to find the target; rather it was popped out from background.

The flow chart shown in Figure 9 explains the working of the adaptive MFD. It may be noted that the adaptation strategies are developed for a cursor control device, not for facilitating physical touch in a touchscreen. The blue boxes indicate system-initiated function while brown boxes indicate user-initiated action.

If the cursor moves in the outer region of the MFD (marked in Figure 9), the nearest neighborhood predictor finds the closest hotspot and activates one of the peripheral large buttons. Activation is indicated by changing background color and enlarging the target 1.5 times its size. The user selects the target by pressing the slew button on the throttle stick of the HOTAS.

If the cursor moved in the inner region of the MFD, users were instructed to press the slew button of the HOTAS when the cursor is on or near one of the small buttons. If the slew button is pressed when the cursor is on the target, it is selected immediately. If the cursor is not on the button, the neighborhood is zoomed in. The user then selects the target by bringing the cursor near the target button and then

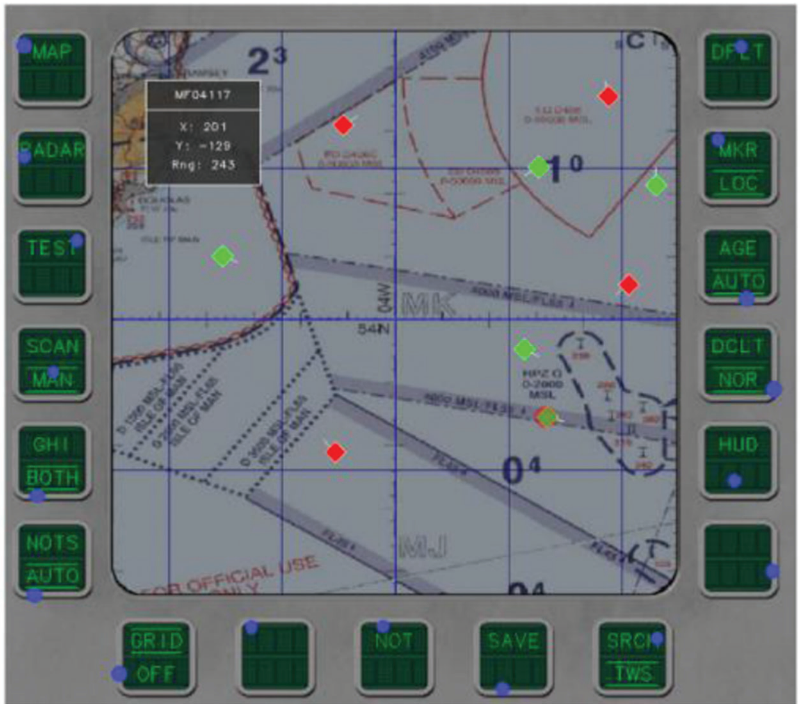


Figure 7. Hotspots for a representative MFD.

a. Small Targets

b. Zoomed-In Position

Figure 8. Zooming for small buttons.

either by pressing the slew button (Adaptation 1) or by staring in (dwelling, Adaptation 2) near the target button.

Eye-Gaze-Controlled HMD

Controlling GUI through wearable eye gaze tracker: We used a screen-mounted gaze tracker and the previous algorithm for controlling a display in HDD and HUD configurations. However, in HMD configuration we used a wearable gaze tracker, as it would not be possible to use a screen-mounted tracker with an HMD. A screen-mounted gaze tracker can specify eye gaze positions in x and y coordinates with respect to a screen. However, a wearable gaze tracker generates x and y coordinates in a normalized form between 0 and 1 covering the entire visual field of an operator. We mapped the normalized x and y coordinates into screen coordinates using a calibration program and a backpropagation neural network. The calibration program displays nine dots on a screen at an interval of 2.5 seconds for displaying each dot. Users were instructed to stare at each dot. We recorded the normalized coordinates while a dot was displayed on screen and stored the median values of the normalized coordinates. After the calibration routine, we generated a file that contains the median values of

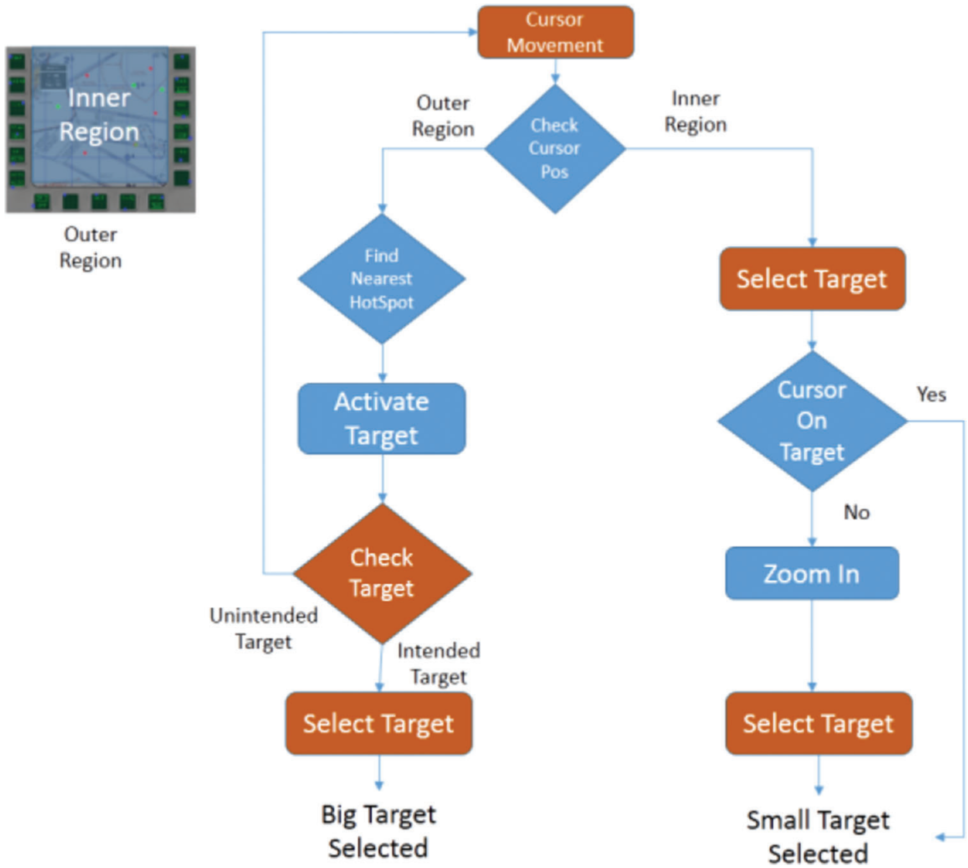


Figure 9. Flow chart of adaptive MFD operation.

normalized coordinates generated by the gaze tracker and screen coordinate of the midpoint of each dot.

We collected data from nine participants using the flight simulator setup shown in Figure 2c and evaluated a linear regression and neural network model for mapping from eye tracker coordinates to screen coordinates. We compared the models by the R^2 and RMS (root mean square) error in prediction. Figure 10 shows the average values and the error bars indicate the standard deviation. Based on the analysis, we noted that the neural network model produces higher R^2 and lower RMS values for error and selected the neural network model (Figure 11).

During subsequent studies with eye tracking glasses, we initially calibrated the tracker by training the neural network. After the calibration, we recorded the normalized coordinates in real time, fed them to the trained network, and drew a cursor on the screen based on the output from the neural network. We recorded the nearest target to the cursor location. If we recorded the same target for seven consecutive times, we activated it.

Study 1: HDD

This study describes a dual task study using a 10″ LCD display for rendering an MFD in head-down configuration.

Participants: We collected data from 12 participants (10 male, 2 female; age range 25 to 35 years, average age 28.3 years). All participants volunteered for the study and did not report any problems with the experimental setup during the practice session. None of them had used a gaze-controlled interface or the particular flight simulator setup before the study. Participants were trained with the flight simulator: they were selected based upon their performance in maintaining altitude and direction in the flight simulator.

Design: The study was designed as 3 × 2 repeated measure design with the following factors:

- Modality: touchscreen × Adaptation 1 × Adaptation 2.
- Button size: large buttons × small buttons.

Participants were required to press large and small buttons alternately on hearing the auditory cue. Each participant was instructed to press 8 large and 8 small buttons alternately. We measured the following dependent variables:

- Flying performance was measured by
  - Deviation from the straight line flying path.
  - Deviation in altitude outside the specified envelope of 1000 and 2000 feet.

Figure 10. Comparing linear regression and neural network models for predicting screen coordinates from eye tracking glasses.

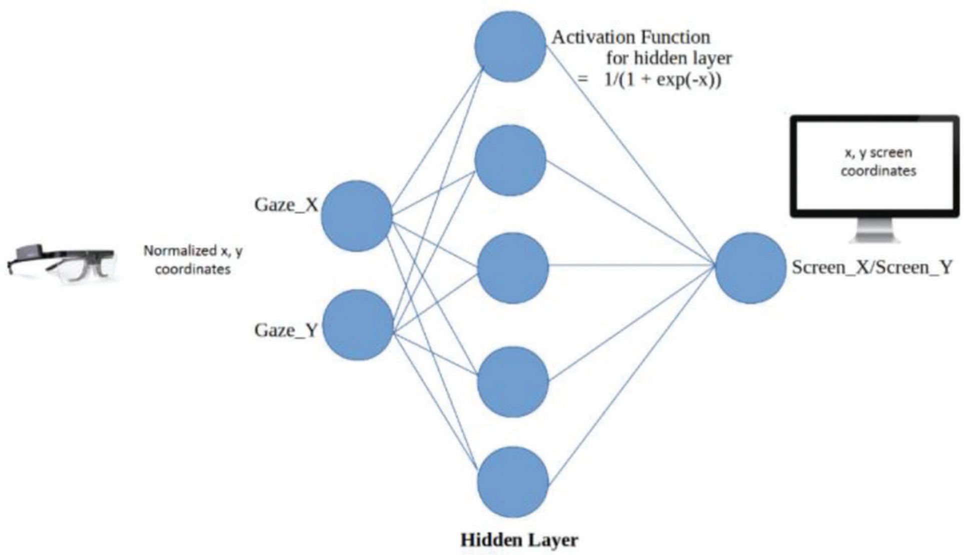


Figure 11. Neural network for using eye tracking glass as a cursor control device.

- Pointing and clicking performance was measured as
  - Response time as the time difference between the auditory cue and the time instant of the selection of the target button. This time duration adds up the time to react to auditory cue, switch from primary to secondary task, and the pointing and selection time in the secondary task.
  - Error in secondary task as the number of wrong buttons selected.
- Cognitive load measured using the NASA Task Load Index (TLX) score.
- Subjective preference as measured by the System Usability Score (SUS) (SUS, 2014).

Procedure: Initially participants were briefed about the purpose of the study. Then they were trained with the flying and pointing and selection tasks separately and undertook both tasks separately. We proceeded with the trial only when they could complete each task separately. Finally, they undertook the dual task study. In this study, we used a clear visor of the helmet, while a later study separately collected data on clear and dark visors. On completion of each condition, we recorded the cognitive load and preference using TLX and SUS questionnaire.

Results: All participants completed the trial in all conditions. In total, we recorded 580 pointing tasks, although they did not select an equal number of buttons in each condition. In the subsequent graphs, error bars represent standard deviation.

For response time calculation (Figure 12), initially we removed outliers by removing values which are greater than outer fence (Q3 + 3 $\times$ interquartile distance) and discarded 11 pointing tasks. For the remaining tasks, we calculated average response time for large and small buttons separately for each participant and undertook a 3 $\times$ 2 repeated measure ANOVA.

The Mauchy test confirmed no violation from the sphericity assumption. We found a

- significant main effect of button size [$F(1,11) = 31.84$, $p < 0.01$, $\eta^2 = 0.09$] and
- significant interaction effect between button size and modality [$F(2,22) = 11.54$, $p < 0.01$, $\eta^2 = 0.51$].

The main effect of modality was not significant.

A set of pairwise comparisons using *t*-tests among different conditions only found that the response times for small button selections were significantly slower for the first adaptive condition compared to touchscreen.

We also conducted a set of pairwise unequal variance *t*-tests on the response times without averaging them for individual participants and found:

- A significant reduction in response time for selecting large peripheral buttons for the first adaptive condition and considering both adaptive conditions together compared to touchscreen [$t(0,232) = 2.53$, $p < 0.05$ for first adaptive condition; not significant for second condition alone, $t(0,284) = 2.07$, $p < 0.05$ considering both conditions as both conditions used similar technique for large buttons].
- A significant increase in response time for selecting small buttons in both adaptive conditions compared to touchscreen [$t(0,80) = 3.24, p < 0.001$ for first adaptive condition, $t(0,127) = 2.14$, $p < 0.05$ for second adaptive condition]. The second adaptive condition reduced the average response time by 9% compared to the first adaptive condition.

Finally, we combined the response times for large and small buttons together (Figure 13) and evaluated the overall response times. This analysis aims to evaluate whether increases in response times for small buttons are great enough to make a significant difference in overall response time. We did not find any significant difference in response times in a one-way ANOVA with the overall response time being lowest for the first adaptive condition and highest for the second adaptive condition.

There was only one instance of wrong selection for the second adaptive condition. However, it may be noted that for small button selection in adaptive condition, participants were instructed to click anywhere near the target area to zoom in and so click action near the area (but not exactly on the target button) was not considered as a wrong selection.

We did not find a significant difference for any other dependent variables, although the deviation from flight path was lowest in the first adaptive condition while the cognitive load was 10% lower and preference 20% higher for the second adaptive system compared to touchscreen (Figure 14).

Discussion: This study found that the gaze-controlled interface reduced response time for larger peripheral buttons of an MFD, while smaller buttons were still quicker to select using touchscreen. Use of the gaze-controlled interface did not degrade flying performance; rather it was best in one of the adaptive conditions and users' subjective ratings about cognitive load and preference also favored the gaze-controlled interface over touchscreen. The overall response time was not different across three conditions, the lowest being for the adaptive condition. It may also be noted that all users used touchscreen everyday while they were using a

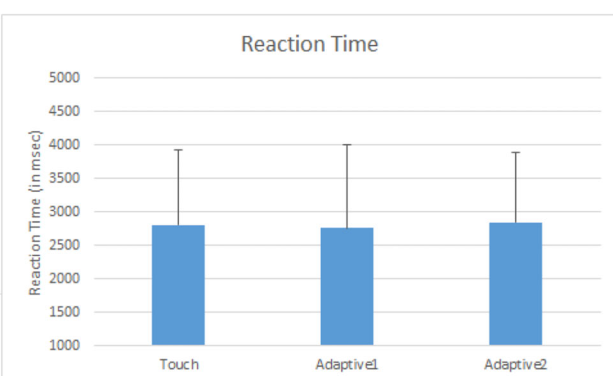


Figure 12. Comparing response times.

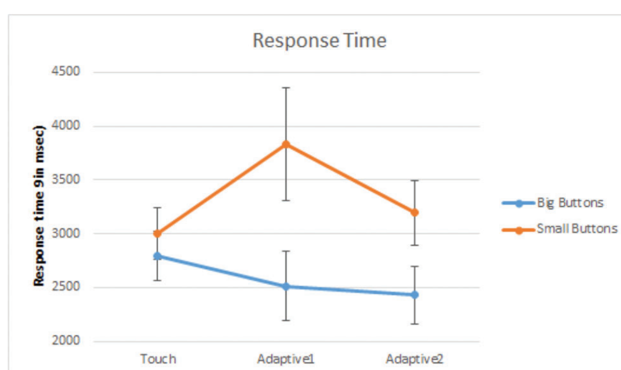


Figure 13. Response times combining large and small buttons.

gaze-controlled interface for the first time. In a high-G environment, it is difficult to move one's hand. Pilots will still use saccadic eye gaze movement to watch MFDs and our gaze-controlled interface can leverage this gaze movement to directly control MFDs. Although users exhibited significantly longer response time for small buttons in the adaptive conditions, the one with dwell time selection (Adaptation 2) reduced pointing and selection time compared to the other adaptive condition (Adaptation 1) by 100 ms on average. Future studies may adapt and optimize dwell time duration to further reduce response times. In the following studies, we explored the gaze-controlled interface as an HUD.

The following study rendered an MFD on a projected display configured as an HUD and evaluated its performance. We undertook a pointing and selection task similar to this study and compared performance of the gaze-controlled interface with respect to the standard TDS, presently used to operate both HDDs and HMDs.

Study 2: HUD

This study involved similar tasks to the first study. However, instead of analyzing large and small buttons separately, we compared the gaze-controlled interface with different visors of the helmet in a projected display.

Participants: We collected data from 11 participants (9 male, 2 female; age range 25 to 34 years) recruited from our university with similar sampling criteria to the previous study discussed above. The participants had no visual, cognitive, and motor impairments and did not have any problem in using the experimental setup. None of them had used either a HOTAS joystick or eye gaze tracking-based interface earlier.

Design and Material: The study used a similar set of materials to the previous studies. We used the screenshot of Figure 6 and each participant was instructed to make at least eight selections for each condition while undertaking the flying task. We also used a helmet given to us by HAL. The helmet had clear and dark visors. Since the eye gaze tracker works based on reflected infrared light, we wanted to evaluate and compare the quality of interaction with a dark visor. Our study consisted of the following three conditions for pointing and selection tasks:

a. Using HOTAS joystick.
b. Using eye gaze tracking with clear visor (ETC).
c. Using eye gaze tracking with dark visor (ETD).

Procedure: We followed the same procedure as the first study but recorded data separately with clear and dark visors.

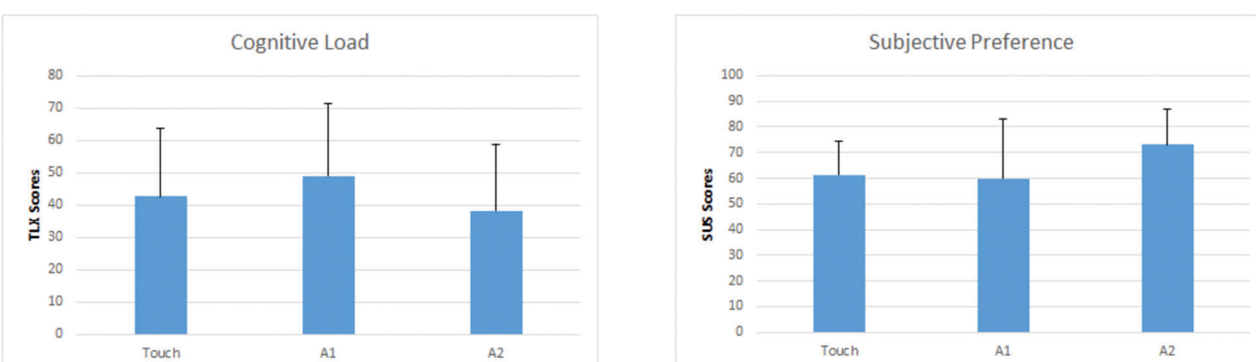
Results: We initially analyzed the flight performance under three different conditions in terms of the standard deviation in the *x*-axis, standard deviation in altitude when the flight was outside the envelope between 1000 and 2000 feet, and total distance covered during the task. We did not find any significant difference among those parameters in one-way ANOVAs. In fact, the deviation in altitude (Figure 15) was 10% less with eye gaze tracking with the clear visor compared to the HOTAS joystick.

Next we analyzed response times for 268 pointing tasks and found a significant difference among the three different conditions [$F(2,246) = 9.65$, $p < 0.05$, $\eta^2 = 0.07$]. A set of pairwise *t*-tests also found significant difference between the response times for the gaze-tracking system and the HOTAS joystick (Figure 16).

The number of wrong selections was measured and reported in terms of percent error, which was lowest for the joystick and was under 10% for all conditions. There were four wrong selections for HOTAS, nine for eye gaze tracking with the clear visor and six for eye gaze tracking with the dark visor.

Users reported significantly less cognitive load (Figure 17) for gaze-controlled systems in terms of TLX scores [$F(2,26)= 5.04$, $p < 0.05$, $\eta^2 = 0.28$]. They also reported significantly higher preference [$F(2,26) = 8.78$, $p < 0.05$, $\eta^2 = 0.40$] for the gaze-controlled system over the HOTAS joystick in terms of SUS scores.

Discussion: This study demonstrates that for the standard flying task, the eye-gaze-controlled interface can perform



a. Cognitive Load b. Subjective Preference

Figure 14. Subjective feedback for MFD study with HDD.

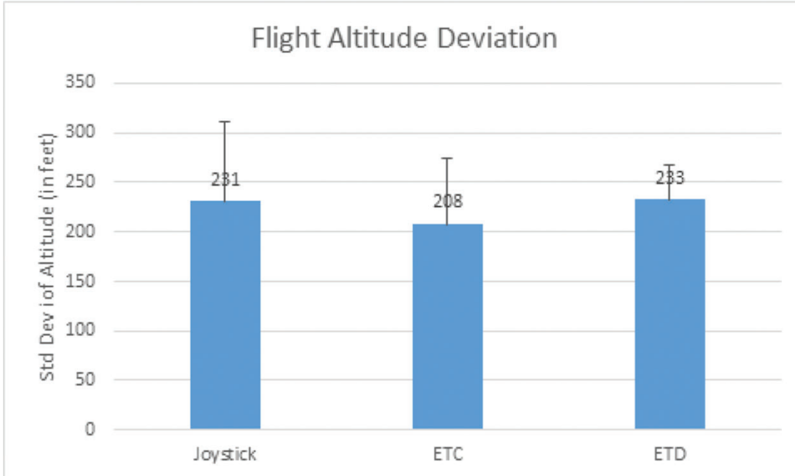


Figure 15. Quality of flying with respect to change in altitude.

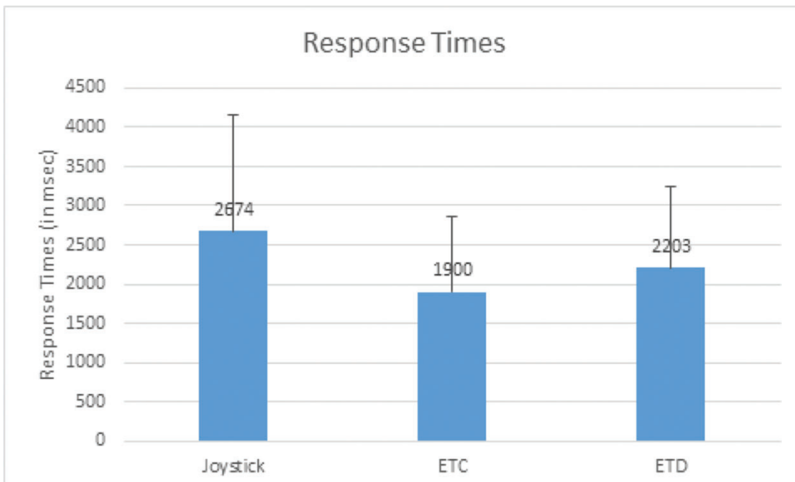


Figure 16. Analyzing response times in flight simulator.

better than the existing HOTAS joystick in terms of both response times and subjective preference. The average response time was even lower than the gaze-controlled HDD reported by Biswas and Langdon (2015) in a single-task study. The flying performance in terms of deviation in altitude marginally improved for the gaze-controlled system compared to the joystick. User preference was also significantly higher for the gaze-controlled system. In summary, we can conclude that even for novice users, the gaze-controlled system can improve performance with secondary mission control tasks in a military aviation environment without degrading the primary flying task.

Study 3: Simulated HMD

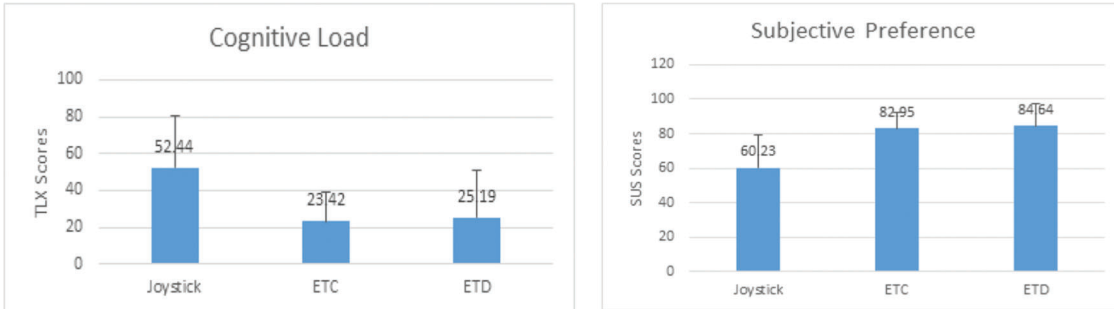
This study setup involved a screen at a sidewise position of the pilot and used a wearable gaze tracker instead of a screen-mounted one. We compared the gaze-controlled system with the TDS as in the previous study.

Participants: We collected data from nine participants (five male, four female; age range 25 to 34 years) recruited from our university with similar sampling criteria to the previous study discussed above. The participants had no visual, cognitive, and motor impairments and did not have any problem in using the experimental setup. As they were required to wear eye-tracking glasses, we only selected participants who did not have glasses or used contact lenses.

Material: We used Tobii Pro Glasses 2 (Tobii, 2018a) for tracking eye gaze and an HP Spectre laptop with Intel i7 core processor to record and process gaze tracking data in real time. The flight simulator used a similar set of materials to the previous studies described above.

Design: Participants were instructed to undertake 10 pointing and selection tasks in parallel with the primary flying task. We compared two conditions:

- Pointing and selection with TDS.
- Pointing and selection with eye gaze tracking.



a. Cognitive Load b. Subjective Preference

Figure 17. Subjective feedback for MFD study with HUD.

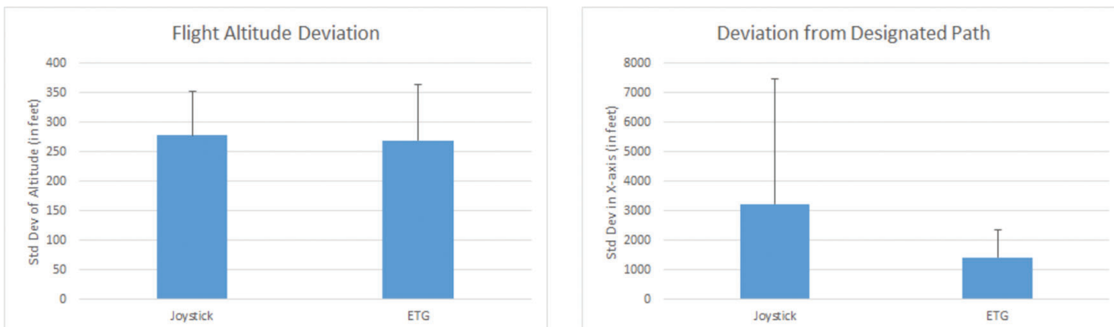


Figure 18. Quality of flying with respect to change in altitude and *x*-axis deviation.

Procedure: We followed similar procedures to those of the previous two studies. The order of using TDS and gaze tracker was randomized to avoid order effect.

Results: We did not find any significant difference in a Wilcoxon signed rank test for primary flight parameters. However, standard deviations in the *x*-axis and in altitude were both lower for the eye tracking system than for TDS (Figure 18).

We found a significant reduction in response times in the eye tracking system compared to the joystick-based TDS [$W = -45$, $p < 0.01$]. All participants could select the target faster using the gaze-tracking system than using TDS. Figure 19 shows the average response times for secondary pointing and selection tasks. It may be noted that all participants could undertake the secondary task faster using the gaze-controlled system than joystick-based TDS.

Finally, we compared the TLX and SUS scores as metrics of cognitive load and subjective preference (Figure 20). The cognitive load measured through TLX was lower in the eye gaze tracking-based system, although the difference was not significant in a Wilcoxon sign rank test. The subjective preference measured through SUS score was significantly higher in the eye gaze tracking system in a Wilcoxon sign rank test [$W = 31$, $p < 0.05$].

Discussion: This study demonstrates that:

- One can use a wearable gaze tracker as a cursor control device.
- One can use a wearable gaze tracker to control a GUI which is not in front of or in the line of sight of an operator.

Existing HMDs used in fighter aircraft require an actuation by the HOTAS switch. Our study aims to eliminate this HOTAS-based actuation by controlling the HMD through eye gaze. Although we did not use a real HMD, the same algorithm can be used as for a real HMD as the relative position of the eyes with respect to the display will not change in an HMD and just one run of the calibration routine will be enough for the neural networks to convert eye gaze coordinates to HMD coordinates. Existing HMDs have a 40° field of view (Collinson, 1996) while our setup worked within an 82° horizontal and 52° vertical field of view. As the study shows, the gaze tracking system can significantly speed up a pointing and selection task compared to the HOTAS joystick.

Study 4: Involving Military Pilots

The previous studies collected data from university students. Senior pilots suggested collecting data from university students, as they thought that any new technology is most likely to be tested first on a training platform with students with little flying knowledge rather than on an established platform with more experienced pilots. We constantly took suggestions from experienced pilots on designing the flight simulator and flying task during the design of the study. However, in the following

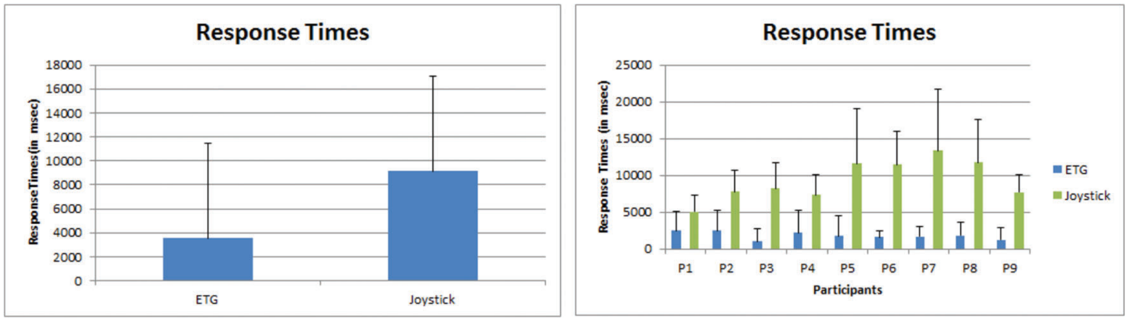


Figure 19. Analyzing response times for simulated HMD.

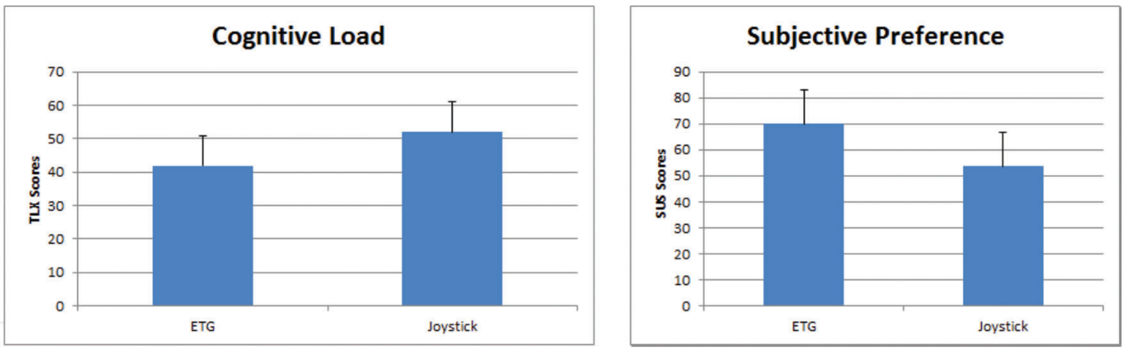


Figure 20. Subjective feedback for simulated HMS study.

subsections, we describe studies involving data collected from military pilots.

We collected data from one pilot of wing commander rank of the Indian Air Force for a gaze-controlled HDD. He could undertake 16 pointing tasks at an average duration of 3.5 seconds (standard deviation = 1.6 seconds) and reported a TLX score of 23 and SUS rating of 82. The average time of selecting large buttons was 2.8 seconds, while that of small buttons was 4.3 seconds. The low TLX score indicated less cognitive load and high SUS score indicated high subjective preference for the system.

In another study described below, we collected data inside an aircraft while flying. We aimed to compare performance of the gaze-controlled system and the proposed nearest neighborhood algorithm that activates the nearest target on screen from the present cursor location.

Participants: We collected data from three IAF pilots with ranks ranging from squadron leader to wing commander.

Material: We collected data using a Microsoft Surface Pro tablet running Windows 10 operating system and a Tobii PC EyeX Minni eye gaze tracker. We used an Avro HL768 transport aircraft for data collection and an X16-1D USB accelerometer from Gulf Coast Data Concepts for recording vibration in units of g ($g = 9.81$ m/s^2).

Design: We set up the tablet and eye gaze tracker at the front seat outside the cockpit as shown in Figure 21.

We used the ISO 9241 pointing task with the following target sizes and distances.

Target sizes (in cm): 1.9, 1.7, 1.5, 1.3, 1.1, 0.9.

Distance of target from center of screen (in cm): 5, 8.

We designed a repeated measure study with the following independent variables:

- *Place of study*
  - On the ground
  - In the air
- *Type of system*
  - Non-adaptive
  - Adaptive, with nearest neighborhood algorithm

We also used an accelerometer in front of the tablet computer to record vibration while flying.

Results: In total, we analyzed 956 pointing tasks with at least 150 tasks recorded for each condition. We calculated average movement time for all combinations of width and distances to target for all different conditions. Figure 22 plots the movement times with respect of indices of difficulties for all four conditions. We found correlation coefficient $r = 0.64$ and $r = 0.63$ between movement time and ID for the non-adapted versions on the ground and in the air respectively. However, with the nearest neighborhood algorithm, the correlation coefficients were less than 0.3.

We undertook a place of study (2) × type of system (2) repeated measure ANOVA on the movement times. We found:

- A significant main effect of place of study $F(1,10) = 14.38$, $p < 0.05$, $\eta^2 = 0.59$.

Figure 21. Aircraft used in the study and placement of setup inside the aircraft.

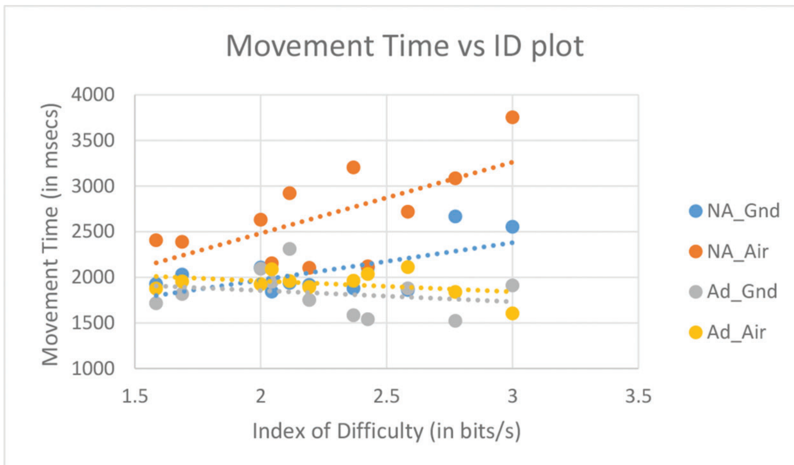


Figure 22. Movement time versus ID plot.

- A significant main effect of type of system $F(1,10) = 34.80$, $p < 0.05$, $\eta^2 = 0.78$.
- An interaction effect of place of study and type of system $F(1,10) = 7.78$, $p < 0.05$, $\eta^2 = 0.44$.

A set of pairwise comparisons found that there are significant differences at $p < 0.05$ in movement times between data collected on the ground and in the air and between adapted and non-adapted conditions on data collected in the air.

In terms of qualitative feedback, all pilots preferred the adaptive version over the non-adaptive one. They noted that the non-adaptive version is difficult to use during takeoff and landing phases compared to cruising phase. In terms of application, they noted that the system will be useful for operating the MFD and operating the HMD for investigating and engaging beyond visual range targets.

We separately analyzed the vibration profile in terms of the acceleration values recorded for roll, pitch, and yaw. The roll and yaw vibration had a maximum value of 1.2G, while the acceleration measured for pitch reached 1.5G.

Discussion: The previous studies were conducted in a laboratory environment with a flight simulator. This study took the system inside an aircraft and measured its performance in vibrating condition. This study shows that the nearest neighborhood algorithm made selection of smaller targets easier, as indicated by the low value of correlation between movement time and ID. This ease of selection of small targets is more useful in the air under vibrating condition than on the ground as indicated by the ANOVA study and pairwise comparisons. It may also be noted that using the nearest neighborhood algorithm, participants can select target using the gaze-controlled interface in less than 2 seconds on average both on the ground and in the air.

However, in this study, we used two different devices for measuring movement time and acceleration, and hence cannot synchronize on the milliseconds level. We could not make separate analyses for different flying phases and, being in a transport aircraft, we could measure performance of the gaze-controlled system up to 1.5G only. In our future studies, we are planning to collect data on a fighter aircraft attaining higher G values and also synchronizing the users' performance with vibration profiles.

Summary

Overall, we can conclude that the gaze-controlled system can improve performance with secondary tasks compared to existing joystick-based TDS. The flying performance was not affected by using the gaze-controlled interface; rather, it was slightly improved in terms of deviation from flight envelope. In all three user studies, users' subjective feedback also favored the gaze-controlled system over touchscreen or TDS. In the following paragraphs we clarified a few issues about the design of our studies and implementation of the adaptation techniques.

- **Interference with caption for hotspots:** When the buttons have captions on them, it may be possible that the optimum position of the hotspot is on the caption itself. It may be noted that the aim of the hotspot is to leverage the pop-out effect and it need not to be a physical spot itself. Bold-facing or rendering in reverse-contrast a single letter of the caption or placing a high-contrast icon can also serve as a hotspot.
- **Adaptable zooming technique:** The zooming technique was adaptable in the sense that the magnification ratio is selected based on the minimum distance between all pairs of targets at any point in time. It was implemented by drawing a window across the required area, which in fact occluded part of the display. However, the occlusion duration was limited by the dwell time duration and was less than or equal to one second in the present implementation. The zooming window disappeared as soon as a gaze location was recorded outside the zooming window. We also implemented a nearest neighborhood predictor within the zooming window so that users need not to bring the cursor exactly on target. Unlike existing work, the magnification did not occur continuously, but rather initiated by pressing the slew button on the throttle. Although we did not find a reduction in pointing times for small MFD buttons compared to touchscreen, use of dwell time reduced reaction times by 9% compared with without it. The zooming technique can also be used with other modalities, and future studies will explore further adaptation options like optimizing the dwell time (Nayyar et al., 2017).
- **Simulated HMD:** We could not use an actual HMD for our study and instead had to use an LCD screen as a simulated HMD. Existing commercial gaze-controlled HMD (Ergoneers 2018; Tobii, 2018b) works as a virtual reality interface, but may reduce situational awareness of pilots. In our particular setup, if we used a virtual reality headset and configured it for a secondary task, participants could not undertake the primary flying task. Existing commercial (Lumus Technologies, 2018) and research work on see-through displays mostly explored near-eye displays (Lewis et al., 2013; Plopski et al., 2015). However, existing HMDs in military aviation, like the Striker ® II system (Striker, 2018), do not use near-eye display; rather, they project the display on a canopy. In our future studies, we plan to set up a pico-projector as an HMD; however, our gaze-control algorithm will work for any display after calibrating it for its relative position with respect to the eyes. It may be noted here that the relative position of an HMD with respect to the eyes will not be affected by head movement and only one calibration will be sufficient for it to work as a gaze-controlled display.
- **Selection for gaze control interface:** In a gaze-controlled interface, selection of a target, popularized as the Midas Touch problem, is not as straightforward as with other interaction techniques such as touchscreen, touchpad, or mouse. A previous study by Biswas and Langdon (2015) explored hardware switch and voice command-based selection while research on assistive technology (Penkar, Lutteroth, & Weber, 2012) explored dwell time-based selection. Considering the timing constraint, we used a button on the throttle (slew button) for selecting a target in the reported user studies, while dwell time was used in one adaptive method and for the simulated HMD.
- **Data analysis strategy:** We checked normality by drawing box plots and comparing mean and median for each individual dependent variable. If the data were normally distributed, we conducted parametric tests; otherwise, we undertook nonparametric tests. For all graphs, the column represents average, while the error bar represents standard deviation.

Conclusion

This paper reports the results of three studies on undertaking pointing and selection tasks with a simulated MFD using eye gaze in addition to flying an aircraft in straight and level maneuver, and compared performance of both flying and pointing tasks with state-of-the-art interaction devices (touchscreen and TDS). We also explored possibilities of rendering an MFD as a HUD instead of traditional HDD configuration. For conducting the study, we designed bespoke hardware as a working prototype of a configurable HUD and designed and implemented new algorithms to control an on-screen cursor through an operator's eye gaze. Using our setup that consists of helmet, gloves, and HOTAS used in existing military aircraft, participants could undertake pointing and selection tasks in an average time of 2.43 seconds (Figure 11) for HDD and 1.9 seconds (Figure 18) for HUD. This time estimation includes context switching from primary flying to secondary pointing task and was significantly better than existing TDS (2.67 seconds) and touchscreen (2.8 seconds). We configured a wearable gaze tracker as a cursor control device using a neural network-based algorithm. Using that, participants could undertake pointing and selection tasks statistically significantly faster than TDS in a side-wise display configured as a simulated HMD. We envision alleviating the need of HOTAS-based actuation for operating an HMD by using an eye-tracking glass. We tested the system inside an aircraft: military pilots could undertake representative pointing and selection tasks using the gaze-controlled interface in less than 2 seconds, on average. The proposed algorithms for the gaze-controlled interface can also be extended to other areas like automotive control and assistive technology.

Acknowledgements

This work was funded by a grant from the Aeronautical Research and Development Board, India.

References

BAE Systems. (2018). BAES 6th generation cockpit. Retrieved from https://www.aerosociety.com/news/wearable-cockpits-the-ultimate-human-machine-interface

Bates, R. (1999). Multimodal eye-based interaction for zoomed target selection on a standard graphical user interface. Proceedings of INTERACT 1999.

Biswas, P., & Langdon, P. (2014). Eye-gaze tracking based interaction in India. *Procedia Computer Science*, *39*, 59–66.

Biswas, P., & Langdon, P. (2015). Multimodal intelligent eye-gaze tracking system. *International Journal of Human-Computer Interaction*, *31*(4), 277–294.

Biswas, P., Prabhakar, G., Rajesh, J., Pandit, K., & Halder, A. (2017). Improving eye gaze controlled car dashboard using simulated annealing. *Proceedings of the 31st British Human Computer Interaction Conference (HCI 17).*

Collinson, R. P. G. (1996). *Introduction to avionics systems*. Dordrecht, The Netherlands: Springer.

de Reus, A. J. C., Zon, R., & Ouwerkerk, R. (2012). Exploring the use of an eye tracker in a helmet mounted display. Avionics Europe Conference & Exhibition, Munich, Germany, March 21–22, 2012.

Dixon, M., Fogarty, J., & Wobbrock, J. (2012). A general-purpose target-aware pointing enhancement using pixel-level analysis of graphical interfaces. *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems* (pp. 3167–3176). New York, NY: ACM.

Ergoneers. (2018). Eye tracking head mounted displays. Retrieved from https://www.ergoneers.com/en/en/hardware/eye-tracking/head-mounted

Eurofighter. (2017). Flight manual for the package Eurofighter Typhoon Professional. Retrieved from https://www.afs-design.de/pdf/AFS_EFpro_English.pdf

Farrell, S., & Zhai, S. (2005). System and method for selectively expanding or contracting a portion of a display using eye-gaze tracking. US Patent 20050047629 A1.

Furness, T. A. (1986, September/October). The super cockpit and its human factors challenges. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 30, pp. 48–52). Dayton, Ohio, USA.

Grandt, M., Pfendler, C., & Mooshage, O. (2013). Empirical comparison of five input devices for anti-air warfare operators. Retrieved from http://dodccrp.org/events/8th_ICCRTS/pdf/035.pdf

Handa, S., & Ebisawa, Y. (2008). Development of head-mounted display with eye-gaze detection function for the severely disabled. Proceedings of the IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems, Istanbul, Turkey.

Hierl, R., Neujahr, H., & Sandl, P. (2012). Military aviation. In M. Stein & P. Sandl (Eds.), *Information ergonomics: A theoretical approach and practical experience in transportation* (pp. 159–195). Berlin, Germany: Springer Science & Business Media.

HOTAS (2017). Retrieved from http://www.thrustmaster.com/products/hotas-warthog

Huey, E. (1908). *The psychology and pedagogy of reading* (Reprint). MIT Press 1968.

Jacob, M., Hurwitz B., & Kamhi, G. (2013). Eye tracking based selective accentuation of portions of a display. WO Patent 2013169237 A1.

Lewis, J. R., Wei, Y., Crocco, R. L., Vaught, B. I., Kipman, A. A. A., & Perez, K. S. (2013). Gaze detection in a see-through, near-eye, mixed reality display. US Patent Application US 2013/0050070 A1.

Lumus Technologies. (2018). Retrieved from https://lumusvision.com/

McGuffin, M. J., & Balakrishnan, R. (2005). Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *ACM Transactions on Computer-Human Interaction*, *4*(12), 388–422.

Murata, A. (1998). Improvement of pointing time by predicting targets in pointing with a PC mouse. *International Journal of Human-Computer Interaction*, *10*(1), 23–32.

Nayyar, A., Dwivedi, U., Ahuja, K., Rajput, N., Nagar, S., & Dey, K. (2017). OptiDwell: Intelligent adjustment of dwell click time. *Proceedings of 22nd International Conference on Intelligent User Interfaces* (pp. 193–204). New York, NY: ACM.

Pasqual, P., & Wobbrock, J. (2014). Mouse pointing endpoint prediction using kinematic template matching. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 743–752). New York, NY: ACM.

Penkar, A. M., Lutteroth, C., & Weber, G. (2012). Designing for the eye: Design parameters for dwell in gaze interaction. *Proceedings of the 24th Australian Computer-Human Interaction Conference* (pp. 479–488). New York, NY: ACM.

Pfeuffer, K., & Gellersen, H. (2016). Gaze and touch interaction on tablets. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (pp. 301–311). New York, NY: ACM.

Plopski, A., Itoh, Y., Nitschke, C., Kiyokawa, K., Klinker, G., & Takemura, H. (2015). Corneal-imaging calibration for optical see-through head-mounted displays. *IEEE Transactions on Visualization and Computer Graphics*, *21*(4), 481–490.

Poitschke, T., Laquai, F., Stamboliev, S., & Rigoll, G. (2011). Gaze-based interaction on multiple displays in an automotive environment. *IEEE International Conference on Systems, Man, and Cybernetics* (pp. 543–548). Piscataway, NJ: IEEE.

Ruiz, J., & Lank, E. (2010). Speeding pointing in tiled widgets: Understanding the effects of target expansion and misprediction. *Proceedings of the 15th International Conference on Intelligent User Interfaces* (pp. 229–238). New York, NY: ACM.

Salomon, D. (2005). *Curves and surfaces for computer graphics*. Berlin, Germany: Springer Verlag.

Striker. (2018). Striker® II Helmet. Retrieved from https://www.baesystems.com/en/product/striker-ii-digital-helmet-mounted-display

SUS. (2014). System Usability Scale. Retrieved from http://en.wikipedia.org/wiki/System_usability_scale

Thomas P. (2018). Performance, characteristics, and error rates of cursor control devices for aircraft cockpit interaction. *International Journal of Human-Computer Studies*, *109*, 41–53.

Thomas, P., Biswas, P., & Langdon, P. (2015). State-of-the-art and future concepts for interaction in aircraft cockpits. In, M. Antona & C. Stephanidis (Eds.), *Universal access in human-computer interaction. Access to interaction* (pp. 538–549). Cham, Switzerland: Springer.

Tobii. (2018a). Tobii Pro Glasses 2. Retrieved from https://www.tobiipro.com/product-listing/tobii-pro-glasses-2/

Tobii. (2018b). Tobii Tech Virtual Reality. Retrieved from https://www.tobii.com/tech/products/vr/

TopGunMilitary. (2013, May). Army AH064 attack helicopter [Video file]. Retrieved from https://www.youtube.com/watch?v=UruW0N2mPsY

Vertegaal, R. (2008). A Fitts law comparison of eye tracking and manual input in the selection of visual targets. *Proceedings of the 10th International Conference on Multimodal Interfaces* (pp. 241–248). New York, NY: ACM. https://doi.org/10.1145/1452392.1452443

Ware, C., & Mikaelian, H. M. (1987). An evaluation of an eye tracker as a device for computer input. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 183–187). New York, NY: ACM.

Wood, R. B., & Howells, P. J. (2001). *Head-up displays: The avionics handbook*. Boca Raton, FL: CRC Press.

Zhai, S., Morimoto, C., & Ihde, S. (1999). Manual and gaze input cascaded (MAGIC) pointing. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 246–253). New York, NY: ACM.

Jeevithashree DV, is a doctoral student from Intelligent Inclusive Interaction Design (I3D) Lab, Centre for Product Design and Manufacturing (CPDM), Indian Institute of Science (IISc), Bangalore. She holds a master's degree in Computer Science & Engineering from VIT University, Vellore and a bachelor's in information science and engineering from PESIT Bangalore. She is currently working on investigating Human Computer Interaction issues in a limited mobility environment for developing intelligent eye gaze-controlled interfaces. Her research mainly focuses on two distinct user groups: children with severe speech and motor impairment (SSMI) and military fast jet pilots.

L. R. D. Murthy, received a B.Tech degree in electrical and electronics engineering from SASTRA University, India in 2015. He is currently pursuing a PhD at the Intelligent Inclusive Interaction Design (I3D) lab in the Centre for Product Design and Manufacturing, Indian Institute of Science. His research interests include multimodal human computer interaction, machine learning, and artificial intelligence.

Kamal Preet Singh Saluja, is a researcher at the Intelligent Inclusive Interaction Design (I3D) lab, Centre for Product Design and Manufacturing (CPDM), Indian Institute of Science (IISc), Bangalore. He earned his first degree in computer science and engineering from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal. He is investigating design constraints and developing eye gaze-controlled interaction systems for children with severe speech and motor impairment. He is also working on assessing cognitive impairment in elderly people.

Pradipta Biswas, is an assistant professor at the Centre for Product Design and Manufacturing of Indian Institute of Science and co-chair of Study Group 9 at the International Telecommunication Union. His research focuses on user modelling and multimodal human-machine interaction for aviation and automotive environments and for assistive technology. Earlier, he was a senior research associate in the Engineering Department, a research fellow at Wolfson College, and a research associate at Trinity Hall of the University of Cambridge. He completed a PhD in computer science at the Rainbow Group of the University of Cambridge Computer Laboratory and Trinity College in 2010, and was awarded a Gates-Cambridge Scholarship in 2006.