Modeling Human-Autonomy Interaction of Drone Pilots in Disaster Environments

by

Ben Welton

Department of Electrical and Computer Engineering
Duke University

Date: ____________________________
Approved:

____
Dr. Mary Cummings, Supervisor

____
[Department Committee Member #1]

____
[Department Committee Member #2]

____
[External Committee Member]

Contents

Acknowledgements

Thank you to Dr. Missy Cummings, Songpo Li, Minwoo Kim, Ted Zhu, and Alex Stimpson for their work, advice, and close mentorship. Thank you also to everyone who helped set up the environment and run the experiments: Kausthub Kunisi, Merle Nye, Suman Bajgain, David Benac, Eddy Li, Yao Yuan, Rebecca Schmidt, Samuel Liu, Mihir Dutta, Justin Havas, and Niek Koerkamp.

1

Introduction

Unmanned aerial vehicles (UAVs) are an interesting case study for research, not least because of a recent explosion in the applications that they can exploit as the technology behind them continues to improve and as interest increases in simultaneous control of multiple UAVs. Falling costs and increased reliability have similarly continued to result in more widespread usage of UAVs. [14] These use-cases have traditionally dealt with outdoor, large-scale traversal and have ranged from military usage for reconnaissance to 3D topology mapping for architectural sites. [6] UAVs also offer an opportunity to navigate through more cluttered indoor spaces that are either difficult for humans to access, dangerous for humans to traverse, or just generally benefit from the introduction of autonomous systems. Research set within these indoor systems has the added benefit of producing substantial amounts of data to be collected from motion capture systems due to the highly accurate position information and, for better and for worse, the lack of external pressures such as wind on the UAV's flight path. Past work within these environments has varied, but it predominantly focuses on multiple UAV swarm behavior or the general feasibility of constructing these environments. [7, 13, 5]

The impetus for the studies done in Duke University's Humans and Autonomy Lab focuses more on understanding how to train operators to utilize the semi-autonomous systems needed to capitalize on sending drones into unfamiliar and inaccessible indoor spaces, such as in the example case of needing to send a drone into a damaged nuclear facility. Designing systems and environments to explore these opportunities is both a chance to better understand how to set up a larger system for commercialized applications of these use cases and an opportunity to better understand how well humans interact with autonomous systems as a whole, particularly for systems that they may have little to no experience with. UAV operators frequently receive smaller amounts of training than traditional pilots and have been found to crash at substantially higher rates. [3] A better understanding of how to train operators will also prove crucial as the types of systems that UAVs are being used in continues to expand to unfamiliar circumstances and interfaces. Efforts to create systems where a single operator manipulates multiple UAVs will especially depend on how well these operators can be trained. [11] There is already a sizeable volume of literature devoted to the question of how to maximize for efficient human use of semi-autonomous systems, as poorly designed systems with increasingly advanced forms of autonomy don't necessarily allow for human exploitation of these technological improvements. Human operators who fail to understand how to use given systems either due to poor design or poor training expose themselves to misuse of these tools, over-reliance on autonomy in cases where they should incorporate manual control, or lack of trust in the tools that they have been given. Similarly, poorly designed systems are likely to expose operators to higher workloads that decrease performance or lead to mode confusion and subsequent human errors. [12] Low familiarity environments face particularly acute challenges, since small problems can lead to dramatic shifts in human behavior.

This paper seeks to examine how these elements of system design influence operator strategy through the lens of two experiments that have so far been run within the space, lending special attention to how the results and observations from these experiments differed from expectation. Participants in each of these two experiments were given two interfaces that offered differing amounts of autonomy in the Levels of Autonomy (LOA) framework. One of these interfaces falls near the lowest extreme of this scale, allowing users to directly control rates of movement from the drone. The other provides a higher level of more knowledge-based autonomy in the framework of Human Supervisory Control (HSC) where the operator intermittently engages with a flight plan that they create, leaving the low-level motion details to the system. [2] In this model, the operator takes on the role of the supervisor in a feedback loop, interacting with the system only when adjustments are necessary. Participants were then asked to fly a drone through an obstacle course after limited trainings with these applications. Understanding the outcomes of these experiments provides information about how to train for UAV applications and can help improve design of these experiments in more complex simulations.

This paper will first present an overview of the system and experimental design used during both experiments. It will then individually present the results found during each of the experiments. Finally, it will consider the implications of these results on the system as a whole and what the knowledge gained across both experiments can reveal about operator behavior and about the strengths and weaknesses of the system design.

2

Methods

In order to examine how important varying types of manual and supervisory trainings are for operator performance, a robust indoor system needed to be developed to test them in. Two experiments were then run in the Humans and Autonomy Lab at Duke University. Both relied upon the same motion capture environment, two custom applications developed to present different levels of autonomy, and a variety of training programs that prepared participants on some mix of the two interfaces.

2.1 System Design

The system used for these experiments relied upon four core components: the drone, the motion capture setup that tracked the physical position of that drone, the human-facing controller that received action commands from the participant, and the flight controller that acted as the central node translating information between the three.

The drone used for these experiments was the Parrot ARDrone2. The ARDrone was a perfect choice for an environment like this because it's age and industry familiarity meant that it was extremely well-documented, while its target position as a cheap toy drone meant that it was extremely reliable and durable to user crashes. Other alternatives–including the Parrot Bebop–were explored due to concerns about the shorter battery life of the ARDrone, but concerns about reliability outweighed those about battery life.

Vicon's camera system was deployed for the motion capture setup, with 30 cameras being used in total to cover the space. The cameras were mounted on fixed steel beams running across the tops of the walls or from the ceiling rafters to ensure that the positions and orientations were not disrupted. Of the 30 cameras, 6 of the longer-range Vantage cameras were used and strategically placed to provide coverage for longer open stretches of the environment, while the 24 remaining Vero cameras were used to fully cover the remaining space. Much care was taken during each experiment to ensure that every part of the space was covered by at least three cameras.

The controller provided to participants was a Lenovo Tab 2 with two custom Android applications running on it. Both applications were based in part off of the open-source PPRZonDroid application developed as a part of the PaparazziUAV flight controller. [10] The architecture for each application is identical and simply has a background thread that processes incoming UDP messages about drone status and navigation from the flight controller and passes back formatted TCP messages whenever the user performs an action. The only direct interaction that the application has with the drone is in acquiring the video feed, which is forwarded as an RTP stream from the drone over the network router to the applications. An additional background logger was also added during the second experiment in an effort to collect more comprehensive data. The logger outputs a constant stream of position information along with any actions that the user takes to a CSV file kept locally on the tablet.

The flight controller used in this system was the same PaparazziUAV controller that the PPRZonDroid application was initially developed for. [8] The flight controller uploads its software to the drone to enable communication between the two over the drone's wifi connection. An app server handles the middle layer of communication between the tablet and the flight controller, taking Paparazzi messages and sending them over a UDP stream to the application. User inputs are passed back to the app server over a TCP stream before being parsed and passed along to Paparazzi.

While the code that controls how the drone actually converts Paparazzi messages into flight commands and how it stabilizes itself remained unchanged, a substantial amount of work went into making sure that this flight controller could communicate with each of the other parts of this system. First, a profile of the ARDrone being used in the experiments was created, specifying the appropriate tuning parameters and settings. Second, a script was created to parse the message packets being received over the connected Ethernet port from Vicon into meaningful position data for the flight controller to substitute in lieu of GPS coordinates. Finally, some custom message formats were added to the Paparazzi app server to enable additional information to be passed from the flight controller to the application.

2.2 Interface Design

Two drone control interfaces were developed for this experiment. A manual control interface was designed to present a lower level of autonomous control with a control scheme similar to a traditional physical hand-held controller for a commercial drone. A supervisory control interface was also designed to present a higher level of autonomous control with waypoint-driven navigation. No changes were made to the application interfaces between the two experiments, though comprehensive logging that recorded operator interactions and a record of drone position was added for the second experiment to aid in data analysis.

2.2.1 Manual Interface

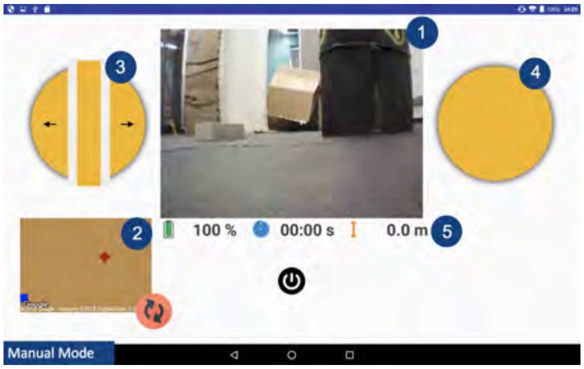
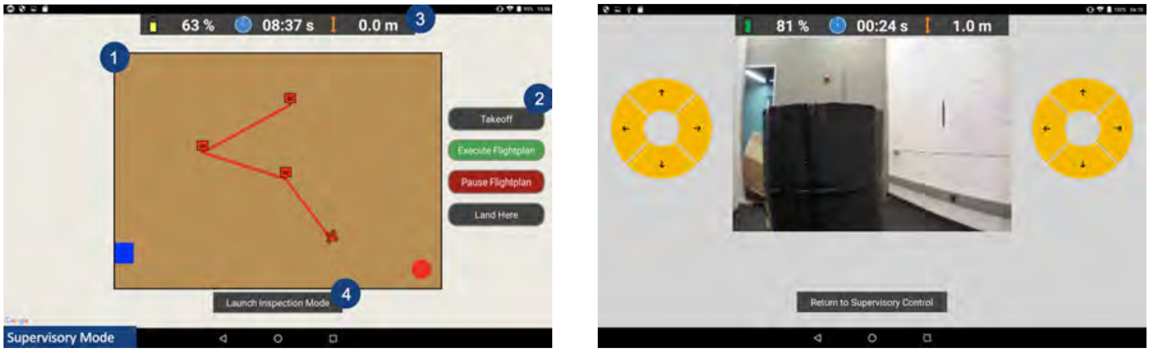


Figure 2.1: Interface for manual control app

The manual control app, seen in Figure 2.1, was designed to incorporate lower levels of automation and leaves much of the control to direct operator inputs via virtual joysticks that control the roll, pitch, yaw, and throttle. There are four key features of the manual interface. First, featured prominently in the center of the screen, is the primary viewport for the operator (1). This element can feature either a real-time camera feed from on-board the drone or a top-down map that shows the orientation and position of the drone relative to the known floorplan. The smaller viewport in the bottom left corner of the interface (2) features whichever of these options is not currently in the center, and that selection can easily be toggled back and forth using the adjacent button. On either side of the viewport are the joysticks (3 & 4), which represent the main form of interaction between the operator and the interface. The left joystick (3) controls throttle and yaw–which control altitude and orientation, respectively–while the right joystick (4) controls lateral position through pitch and roll adjustments. Below the central viewport is a health and status bar (5), which shows updating information about the battery life of the drone, the altitude, and the elapsed flight time. Finally, there is a small power button below the viewport which is used to start and stop the rotors.

2.2.2 Supervisory Interface

Unlike the manual interface, the supervisory app (Figure 2.2) offers a higher degree of autonomy by hiding decisions about how to get between two points and simply having an operator select a series of waypoints to define a flight path. The core interaction that an operator will have with the interface is placing, deleting, and updating waypoints on the central map (1). Once this flight path is satisfactory, an operator would interact with the simple action buttons on the right side of the screen (2), either sending the drone along its path or pausing it when the operator needs to update a flightplan. A health and status bar identical to the manual interface again reports battery life, altitude, and elapsed flight time (3).



(a) Supervisory Mode (b) Inspection Mode

FIGURE 2.2: Interface for supervisory control app

In contrast to the manual interface, there is a second mode (Figure 2.2.b) that the operator can activate (4) when needing to inspect their surroundings, something that is particularly necessary in a simulated disaster zone when the interface has no other way to inspect for abnormalities in the environment. This alternate mode, known as inspection mode, is similar to the manual interface and features a camera feed in the central viewport along with controls on either side. Unlike the manual interface, these controls are not true joysticks and don't have a range of values that the operator can move between, instead operating as virtual thumb pads. This mode was not intended to be used to travel across large distances and was mostly meant to be a nudge control that users can use to inspect their surroundings and adjust the drone on a small scale accordingly.

2.3 Experimental Design

Both experiments completed for this work followed the same format: operators were given a series of training "modules" that familiarized them with the application interfaces and were then asked to complete a timed, final task.

2.3.1 Operator Training

All participants received a training regimen that was split into six modules: UAV Basics, App Interface Training, Takeoff & Landing, General Navigation, Camera Operations, and Emergency Handling. For each module, subjects would review a short slide deck about the app interfaces they would be using, take a brief online test related to what they'd learned, and, if applicable, participate in a hands-on training.

All modules except the first two involved a hands-on training, though the final hands-on task was designated as the "Checkride" and acted as a smaller obstacle course to ensure that subjects could integrate the skills they'd learned into a similar setup to the final experiment. Participants were allowed to move at their own pace, but there was either a limited number of attempts (Experiment 1) or a generous capped amount of time (Experiment 2), at which point they would be forced to move to the next module. Adding a time cap became the preferable mechanism for dealing with these cases, as it ensured that unlucky or inexperienced operators got a similar level of practice with the drone, even after an arbitrary number of mistakes.

Experiment 1 Training

The first experiment was divided into three groups of participants. Group 1 was given training only on the manual control interface, Group 2 was given training on both the supervisory and manual control interfaces, and Group 3 was given training on only the supervisory control interface.

In the first experiment, the hands-on trainings were identical for every group, but Group 1 was asked to complete them with manual control, Group 2 was asked to complete them once with each interface, and Group 3 was asked to complete them with supervisory control. Table 2.1 gives an overview of what task was required for each module.

Table 2.1: Breakdown of tasks for each hands-on training in Experiment 1

Module	Training Task
1	None
2	None
3	Takeoff, hover in place, then land
4	Navigate around barrel
5	Navigate to and read from control panel
6	Checkride

Experiment 2 Training

The second experiment was divided into four groups that were designed with the results of the first experiment in mind. There was an unexpected negative relationship between the increased training for participants exposed to both interfaces and their performance relative to those that only received supervisory training, which will be discussed in further detail in the Results and Discussion sections [4, 14]. As a result, the new Groups 1 and 2 replicated much of the setup from the first experiment, training operators with both the manual and supervisory interfaces, just in different orders. Group 3 participants were given training in only the supervisory interface. Group 4 participants were given an extended supervisory training known as ”Supervisory Plus” that added additional hands-on training and some general suggestions about optimal strategies for using the interface, particularly in emergency situations.

The hands-on trainings followed a similar format to the first experiment. Tasks were identical for each group. Groups 1 & 2 were asked to complete each task once with each interface (though the order of which task was given first depended on the group), and Groups 3 & 4 were asked to complete each task with supervisory control only. However, participants in Group 4 were additionally asked to complete a slight variation on each task to provide more hands-on experience. These secondary tasks were chosen thoughtfully, encouraging participants to get more general experience and get a better sense of the differences between inspection mode and traditional waypoint control. Table 2.2 summarizes these hands-on tasks and provides details about what they entailed.

Table 2.2: Breakdown of hands-on tasks for each module in Experiment 2

Module	Training Task	Additional Training Task (G4 Only)
1	None	None
2	None	None
3	Takeoff, hover in place, then land	None
4	Navigate in circle around room	Navigate in figure-8 around room
5	Navigate to and read from control panel	Repeat only with inspection mode
6	Checkride	Repeat only with inspection mode

2.3.2 Experiment Trial

Once the Checkride was complete, participants began the final experiment. The manual only group (Group 1, Experiment 1) was asked to complete the final trial with the manual interface; all other groups completed the final trial with the supervisory interface. The interface displayed a simple floorplan, and subjects were given the task of reaching a control panel on the opposite side of the room, reading off a sequence of colors (Figure 2.3), and then returning back to the start point. Additional unmarked obstacles were added to the space, ensuring that successful participants would need to interact both with inspection mode and waypoint control. This experiment was run only once for the first experiment but was run three times whenever time permitted during the second experiment. The course was not changed between trials.

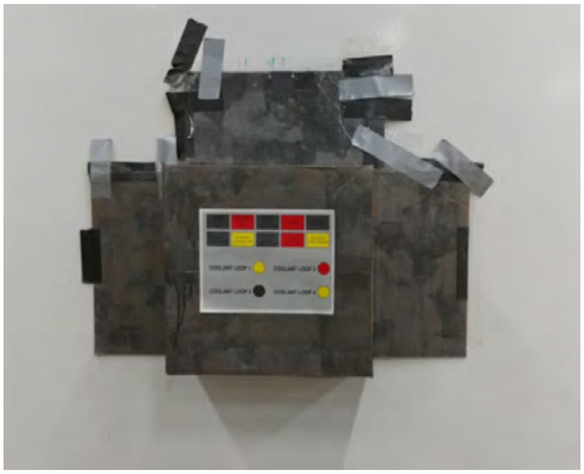
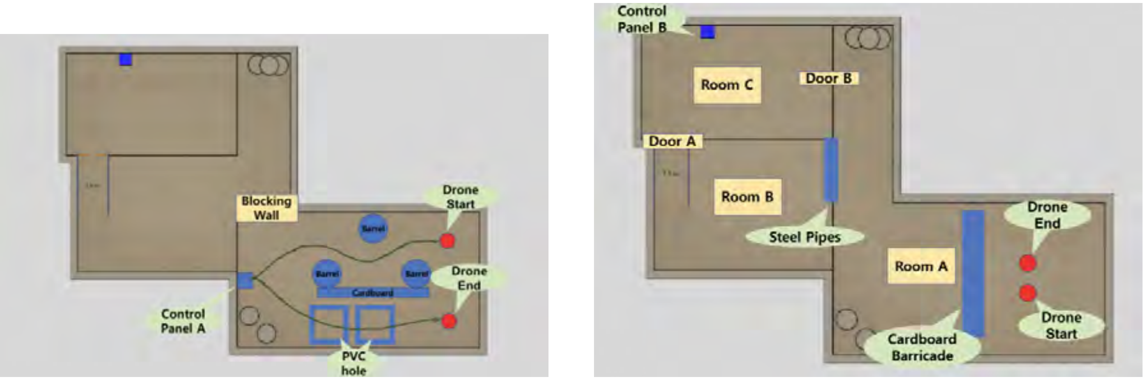


FIGURE 2.3: Colored sequence read off by operators during the final experiment

2.3.3 Environment Design

The general layout and obstacle design of the environment remained largely the same between the two experiments, though the second experiment had a much smaller scale.

Experiment 1 Environment

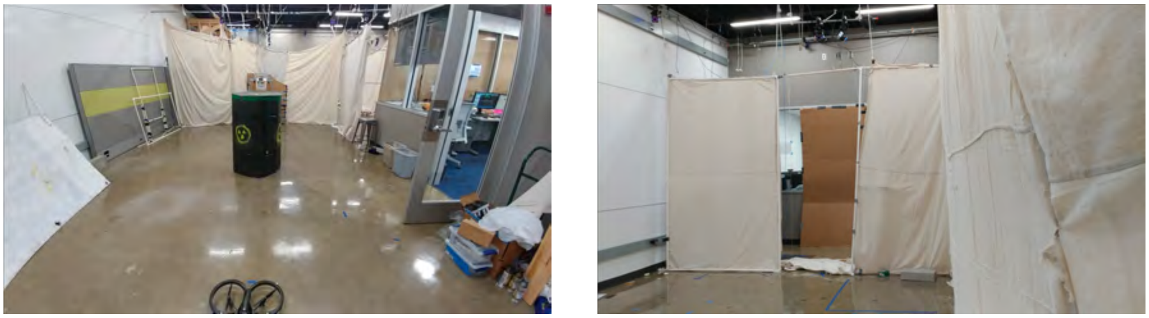


(a) Hands-on Space

(b) Experiment Setup

FIGURE 2.4: Room overview for Experiment 1

Figure 2.4 shows an overview of the room design that was used for the Checkride and for the final experiment. All other hands-on trainings used the same space as the Checkride, just with differing obstacles depending on what module the operator was currently on. For the final experiment, the space was divided using temporary walls made of cloth hung from steel beams. Obstacles included long wooden bars

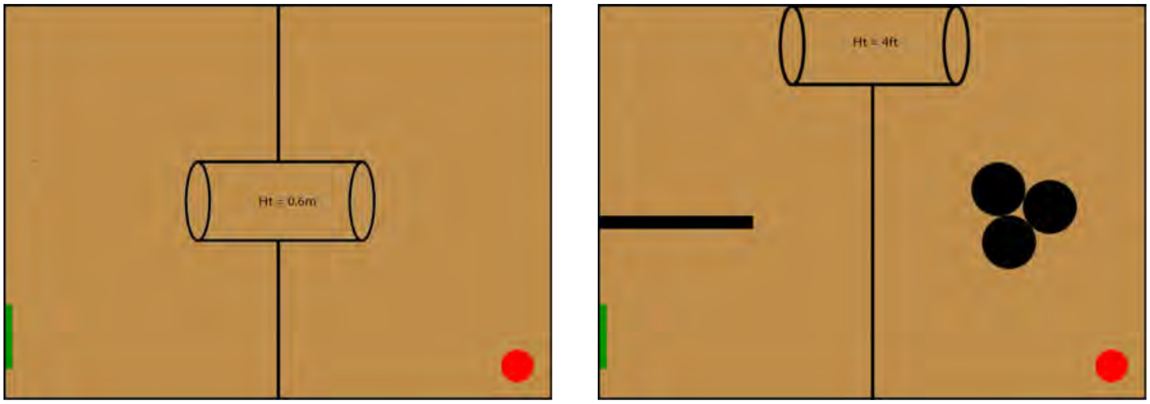


(a) Hands-on Space (b) Experiment Space - Room C

Figure 2.5: Constructed environment for Experiment 1

used as ”steel pipes” to partially block entrance into a room, a cardboard barricade, PVC hoops, a ”ventilation shaft” tunnel, and several ”radioactive” barrels (Figure 2.5). There were also two dynamic elements to the course; when the drone entered Room C through Door A during the experiment, Door A would close and Door B would open for the first time, forcing additional use of inspection mode to maintain situational awareness. Upon arriving in Room C, the operator was asked to read a sequence of colors off a control panel. In order to be considered as a successful trial, the operator then had to fly the drone back through the tunnel and land at the starting point.

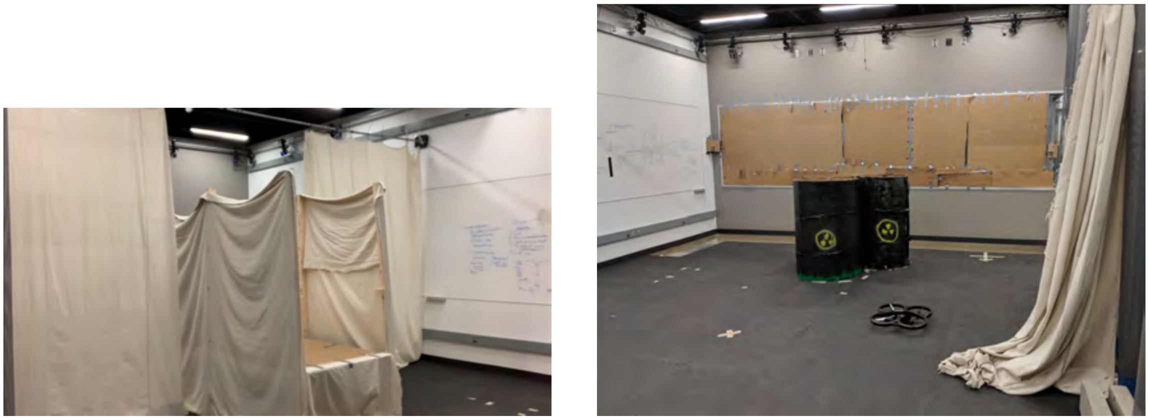
Experiment 2 Environment



(a) Checkride Space (b) Experiment Setup

Figure 2.6: Room overview for Experiment 2

Figure 2.6 shows an overview of the space that was used during Experiment 2. Similar to the first experiment, walls were used to divide the space into subsections, and ”radioactive barrels” were used as additional obstacles. The drone was forced to



(a) Checkride Setup (b) Training Room and Sample Obstacles

Figure 2.7: Constructed environment for Experiment 2

travel through a small tunnel that represented a ventilation shaft in order to arrive at the other side. The second half of the room had a wall that the drone had to navigate around in order to arrive at a control panel that the operator was required to read information from. Figure 2.7 shows parts of the actual environment in more detail. A successful trial was registered if the operator could fly back through the tunnel and return to the starting point. Unlike the previous iteration of this experiment, the overall scale of this space was comparatively small (1/3 of the size of Experiment 1), and it featured no dynamic obstacles that changed depending on how far the operator was in the course.

3
Results

For all statistical analysis in this section, a significance level of $\alpha = 0.05$ was used. Particular attention was focused towards the second experiment, since Zhou et. al and Kim already assessed the general results from the first experiment in great depth and because more data was collected for the second experiment. [14, 4]. Note that manual control training is abbreviated as MC, supervisory control training as SC, and supervisory control plus training as SC+.

3.1 Experiment 1 Results

In Experiment 1, only 34 of the 38 participants were included as data points due to system errors in the remaining 4.

3.1.1 Success Rates

Table 3.1 shows the pass/fail results for participants in the first experiment. Unexpectedly, there was actually a lower success rate for operators who were exposed to both types of training, as opposed to those who only received supervisory training. However, no statistical difference in the success rate between any pairing of groups could be verified when running Fisher's exact test, probably in part because the sample size was small.

Table 3.1: Details on the success rate for each group in Experiment 1

	MC (G1)	MC & SC (G2)	SC (G3)
Success Count	5	3	6
Failure Count	5	5	2
Total Attempts	10	8	8
Success Rate	.50	.38	.75

3.1.2 Completion Times

Table 3.2 shows the average completion times for the first experiment. Unexpectedly, completion times for operators who were exposed to both types of training were not significantly different from those who only received supervisory training. This indicated that some component of the additional training in the second group might actually be lowering performance and provided much of the impetus for the changes in the second experiment.

Table 3.2: Details on the completion times for each group in Experiment 1

	MC (G1)	MC & SC (G2)	SC (G3)
Median (min)	8.22	6.70	7.17
Mean (min)	7.82	7.35	6.97
Std. Deviation (min)	2.17	2.37	1.08
# of Successes	5	3	6

3.1.3 Demographic Factors

Participants in the first experiment had an age range between 21 and 41. Of the 34 total participants, 27 were male and 7 were female. Subjects were overwhelmingly students (31/34), with 29 graduate students and 2 undergraduate students. Very limited data analysis was done on how these demographic factors related to operator outcomes, so much of the discussion here is left to the second experiment.

3.1.4 Feedback and Post-Experiment Results

Once the first experiment concluded, operators were asked to fill out a very brief post-experiment survey assessing what posed the biggest challenge during the experiment. Of the participants included, over a third (11/31) mentioned the camera latency as the most challenging feature of the interfaces that they used. Several others cited the controls as being too sensitive, which caused over-correcting when turning, though this was probably also a result of the camera latency. An assortment of other challenges were also echoed by one or two participants, including a perceived lack of accuracy in the position of the drone on the map, a lack of understanding of orientation of the drone, and a desire to have the camera feed visible on the waypoint-control interface. When asked what was the most challenging part of the final experiment, most participants reiterated their concerns with the camera delay. Additionally, five participants cited a lack of spatial awareness about the amount of space around the drone, and four participants mentioned navigating through the tunnel as particularly difficult.

3.2 Experiment 2 Results

In Experiment 2, only 42 of the 46 participants were included as data points, again due to system errors in the remaining 4. In the interest of collecting as much data as possible, Experiment 2 also asked participants to attempt the final experiment three times when time permitted. As a result, much of the analysis here first examines the initial trial for all 42 participants who attempted it and then the data for all three attempted trials for the 33 participants that had time for three attempts. Note that these participants were asked to repeat the trial three times regardless of their success in each previous trial.

3.2.1 Success Rates

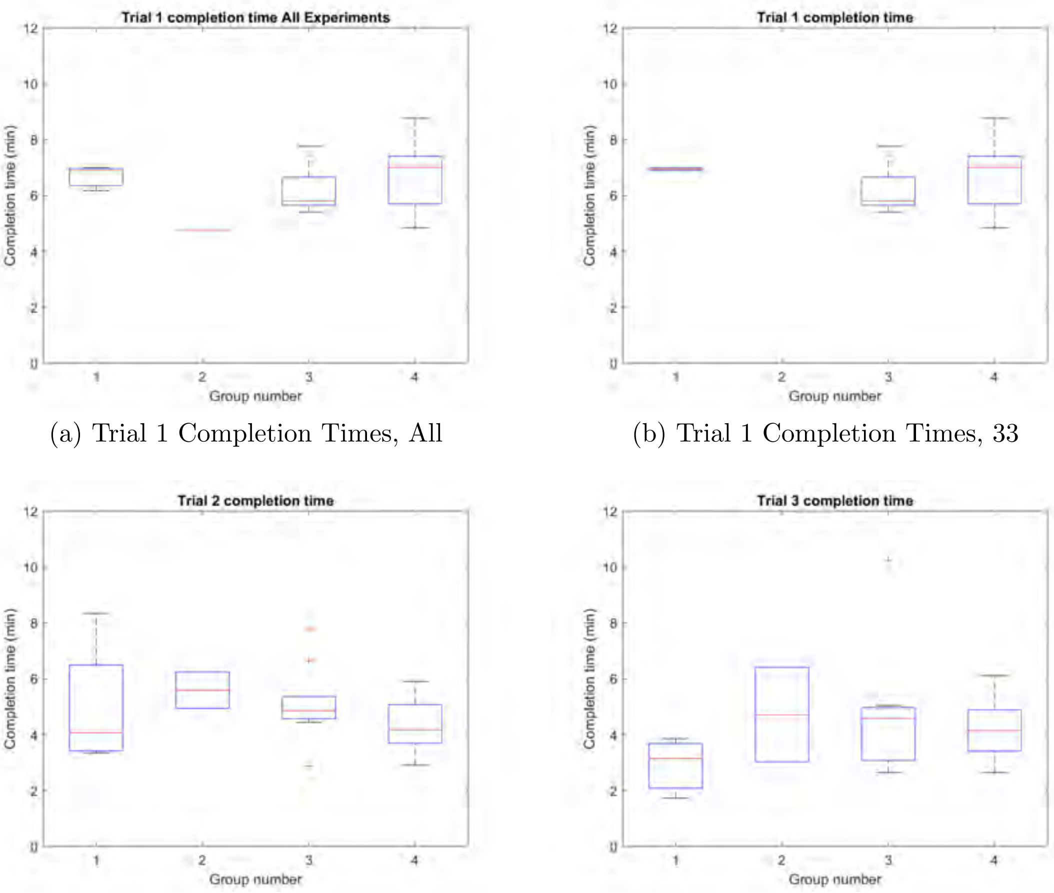
Table 3.3 shows the pass/fail results for participants in the second experiment. Participants exposed to the supervisory plus training had the highest overall success rate. However, no statistical difference in the success rate between any pairing of groups was found when running Fisher's exact test.

Table 3.3: Details on the success rate for each group in the first trial of Experiment 2

	MC & SC (G1)	SC & MC (G2)	SC (G3)	SC+ (G4)
Success Count	3	1	5	7
Failure Count	4	6	9	7
Total Attempts	7	7	14	14
Success Rate	.43	.14	.36	.5

3.2.2 Completion Times

Figure 3.1 shows the boxplot distributions for the completion times for each group in the second experiment, again divided between the 42 participants who completed the first trial and the 33 participants who attempted three trials. One-way ANOVA tests for each of these trials showed that there was no statistically significant difference in the average completion times between any group, though the groups receiving both manual and supervisory trainings had such low numbers of successes to run analysis on that it is difficult to derive much meaning from this. Unsurprisingly, the mean completion times fell for every group with each additional trial as participants became familiar with the course; similarly, the largest drop took place between the first and second trials, since the benefit of seeing a course multiple times should be expected to have decreasing marginal returns.



(a) Trial 1 Completion Times, All

(b) Trial 1 Completion Times, 33

(c) Trial 2 Completion Times, 33

(d) Trial 3 Completion Times, 33

Figure 3.1: Experiment Completion Times

3.2.3 Training Times

Analysis of the training times is not as informative as other metrics, since trainings were intentionally different across groups, and both total training time and hands-on training time should vary substantially. However, there are at least two useful

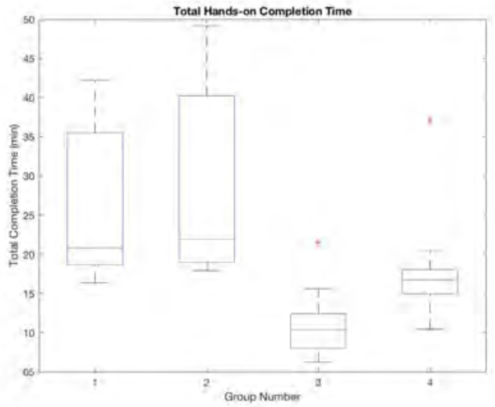


FIGURE 3.2: Total completion times for the hands-on trainings by group

pieces of data that can be extracted from the second experiment. First, variance in training times between the two groups that were exposed to both interfaces can partially demonstrate if the ordering of trainings mattered, since that was the only change in their slides and hands-on trainings. One-way ANOVAs comparing these two groups on both the 42-person and 33-person data sets confirmed that there was no statistically significant difference in total training times or hands-on training times. The only exception to this was total training time for Module 3 ($p = 0.013$), but this is likely a result of the compounding effect of small sample sizes and an exceptionally large outlier in one of the groups. Thus, the ordering of the exposure to manual and supervisory did not influence the time it took to complete the trainings.

One other interesting result arises out of the total hands-on training time, seen in Figure 3.2. Unexpectedly, there is no statistically significant difference in total hands-on training time ($p = 0.1228$) between the group receiving SC training and SC+ training, likely due to the wide range of variant training times for participants in the each of these groups.

3.2.4 Demographic Factors

Of the 42 participants involved in the second experiment, the median age was 22.5, with a range between 19 and 64. 28 participants were male, while 14 were female. Again, subjects were overwhelmingly students (37/42), with 13 undergraduate students, 17 graduate students, and 7 PhD students. Of these students, a large majority came from the Electrical and Computer Engineering or Mechanical Engineering departments. Subjects were predominantly from either the US, China, or India.

Female participants had a slightly higher average completion time across all groups for those that completed the first experiment trial successfully (6.80 min) as opposed to males (6.32 min), but a one-way ANOVA did not find any statistically significant difference between these times ($p = 0.452$). Females did have a lower overall success rate (0.308 vs 0.414), and this trend persisted across all trials completed by the 33 participants that participated in three trials (0.515 vs 0.606).

Education levels similarly did not demonstrate a clear statistical difference in average completion times ($p = 0.632$), though the sample size of PhD students and non-students who completed the trial successfully was quite small (only two successes for each), making it difficult to draw meaningful conclusions.

One of the few demographic factors that did demonstrate a statistically significant difference was previous experience with drones. There was a significantly lower average completion time for subjects that reported any previous drone experience both for the 42 participants that completed the first trial ($p = 0.019$) and across all successful trials for the 33 participants that participated in three trials ($p = 0.024$).

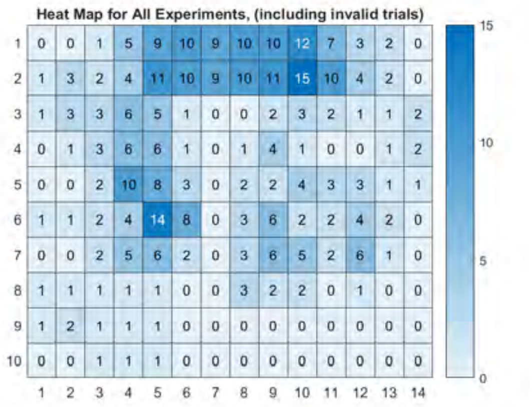
Participants were also polled on several other factors, including how comfortable they felt using remote controllers, how comfortable they felt using tablets, and how frequently they played video games. Somewhat surprisingly, there was no statistically significant difference found in completion times for any of these factors. No clear trends seemed to be present with regards to these demographic factors, though both the 42 participant and 33 participant group had higher success rates for those who reported being relatively uncomfortable with remote controllers as compared to those who reported being relatively comfortable (0.400 vs 0.364 & 0.646 vs 0.510, respectively), possibly indicating that overconfidence with the inspection mode interface might have impacted success rate to some degree.

3.2.5 Feedback and Post-Experiment Results

Once the second experiment concluded, operators were asked to fill out a longer post-experiment survey about how well-prepared they felt going into the final experiment and what parts of the interface they felt most comfortable with. Of the 42 participants included, 5 said that they preferred to primarily rely on waypoint control, 3 said that they preferred to rely on nudge control, and 34 said they preferred to rely on a mix of both. Interestingly, this does not fit well with notes from experimenters, who noticed that many participants tried to complete the first trial almost exclusively in inspection mode. Further analysis of the logged data to see when participants tended to use these different control modes would be worthwhile.

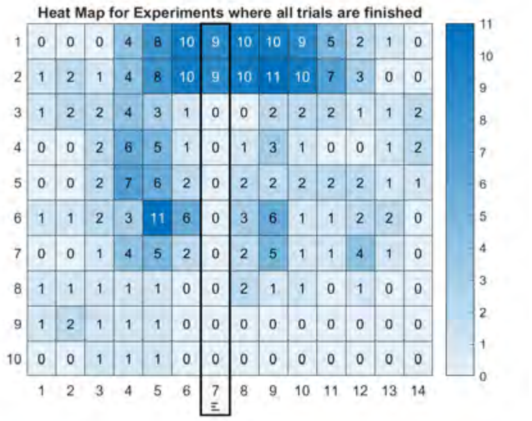
In regards to the training, only 3 out of the 42 participants answered 'No' when asked if the training prepared them sufficiently for the test. When asked to explain, one participant noted that the training was too basic and two participants commented on limited training about spatial awareness and how much space the drone occupied. This latter issue was also echoed by two of the respondents that still felt that the trainings sufficiently prepared them. When asked what participants would change about the training or testing, most participants either reiterated concerns about limited spatial understanding, a desire to have the camera feed visible during execution of a flightplan, or drone instability, particularly at the edges of the walls. Similar concerns were mentioned when asked for any final comments. Five of the participants mentioned delays in the app, specifically with regards to camera latency and the delay between switching modes. Three participants suggested that the drone felt somewhat unstable at times. Three unique respondents also again mentioned uncertainty about the size of the drone and distance to its surroundings in this sections.

Participants were also given a map of the experiment setup and asked to label any areas that they found to be particularly challenging. Figures 3.3 and 3.4 show a heat map based off of this data both for all 42 participants and for the 33 participants that attempted all three trials.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	1	5	9	10	9	10	10	12	7	3	2	0
2	1	3	2	4	11	10	9	10	11	15	10	4	2	0
3	1	3	3	6	5	1	0	0	2	3	2	1	1	2
4	0	1	3	6	6	1	0	1	4	1	0	0	1	2
5	0	0	2	10	8	3	0	2	2	4	3	3	1	1
6	1	1	2	4	14	8	0	3	6	2	2	4	2	0
7	0	0	2	5	6	2	0	3	6	5	2	6	1	0
8	1	1	1	1	1	0	0	3	2	2	0	1	0	0
9	1	2	1	1	1	0	0	0	0	0	0	0	0	0
10	0	0	1	1	1	0	0	0	0	0	0	0	0	0

Figure 3.3: Heat map showing regions that participants self identified to be challenging



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0	0	4	8	10	9	10	10	9	5	2	1	0
2	1	2	1	4	8	10	9	10	11	10	7	3	0	0
3	1	2	2	4	3	1	0	0	2	2	2	1	1	2
4	0	0	2	6	5	1	0	1	3	1	0	0	1	2
5	0	0	2	7	6	2	0	2	2	2	2	2	1	1
6	1	1	2	3	11	6	0	3	6	1	1	2	2	0
7	0	0	1	4	5	2	0	2	5	1	1	4	1	0
8	1	1	1	1	1	0	0	2	1	1	0	1	0	0
9	1	2	1	1	1	0	0	0	0	0	0	0	0	0
10	0	0	1	1	1	0	0	0	0	0	0	0	0	0

Figure 3.4: Heat map showing regions that participants who attempted all three trials self identified to be challenging

4
Discussion

Despite the intuition that more training should improve overall performance, operators across both experiments did not seem to validate this hypothesis. Participants who were exposed to only the supervisory interface trainings managed to have similar success rates and completion times both to those who additionally received manual interface training and to those who received the more specialized supervisory plus training. Certainly, the participants using only the lower autonomy manual control were unequivocally the least efficient operators. Beyond identifying that some autonomy was helpful in improving performance, however, it is not immediately clear that the remaining regimens were at all helpful to operators.

The inability of these extended trainings to produce superior performance certainly doesn't mean that there is nothing to gain from these experiments. In addition to producing a large body of data on operator actions, this work can help answer the question of either what content was missing from these trainings or what unintentionally detrimental behaviors were encouraged by them. Beyond the discussion here, further work applying the Hidden Markov Modeling techniques performed on the first experiment to the second data set will reveal a more detailed understanding of operator strategies and help illuminate the role of trust and how different groups evaluated when to use inspection mode versus waypoint control. [14] System design and interface improvements are also incredibly important in creating stronger human interactions with semi-autonomous systems and are difficult to apply well without having gone through several iterations. In this regard, this work can hopefully generate better methodology for future experiments and use understandings about why these changes need to be made to better understand what is driving human decision-making.

4.1 Designing Effective Training

In trying to resolve how to develop future trainings so that they actually do change performance, it is helpful to consider what factors influenced the lack of improvement for operators exposed to both supervisory and manual training and the operators exposed to supervisory plus training.

Operators exposed to SC and MC

The initial theory explaining the lack of improvement for operators exposed to both interfaces was that manual control training may teach a reluctance towards–and distrust in–the inspection mode interface, causing an overreliance on waypoint control. Participants additionally trained in MC in the first experiment were observed to spend substantially more time transitioning between inspection mode and waypoint-control mode. These participants used inspection mode most frequently just for the camera feed, relying more exclusively on waypoints to navigate. This possibly increased overall time due to delays lost to mode switching and increased the chance of crashing in narrow regions. This lines up well with feedback of participants who complained about lack of spatial awareness and camera latency: increased time training in manual or inspection mode likely increased exposure to the challenges of a poor conception of how much space the drone occupied and may have driven participants away from using the mode, even when it made the most sense. Repeating HMM analysis on the data set from the second experiment can help determine if the same underuse of inspection mode occurred for participants given MC training and SC training in this experiment.

While this explanation is both increasingly plausible and supported by the data thus far, the attempt to see if ordering of these trainings exacerbates or improves this distrust is more difficult to verify from these results. Very few individuals who were given both trainings successfully completed the course, both for those given MC first and for those given SC first. The hands-on trainings across these two groups did not take significantly different amounts of time, but this doesn't necessarily prove a lack of difference, particularly with such a small sample size.

Operators exposed to SC+

The failure of the SC+ training to improve performance is even more surprising. Even if the additional training slides provided little value, the extra hands-on training seemed likely to double operator's level of experience with the actual interface they used in the final trial. Considering previous drone experience was one of the only demographic factors to have any statistically significant impact on completion times and that completion times dramatically decreased with subsequent attempted trials, it seemed likely that this extra exposure would have been quite useful.

Several factors likely contributed to these results. The first is that the additional hands-on trainings may not have been as helpful as anticipated. For Module 4, participants were given an additional task of doing a figure-8 around the room. This simple task took many participants less than a minute and was not substantially different from the previous task. For Modules 5 and 6, participants were asked to repeat the previous task using only inspection mode. This was intended to highlight how much faster operators could move using waypoints, but it may have provided the same distrust in waypoint control that the MC + SC operators were exposed to: by encouraging so much extra time in the inspection interface, these operators likely gained the same distaste for inspection mode as a viable form of navigation in tight spaces. Further analysis on the data set from the second experiment can verify to what degree this was the case.

The second likely contributor is that the emergency issues that were supposed to be explained were either not well established or were unrelated to the problems that are actually the most difficult to resolve in practice. Advice about how and when to switch to inspection mode remains incredibly crucial but wasn't necessarily clear from the slide deck. It may also be worth explicitly mentioning that inspection mode should not be used for the whole experiment; this would possibly prevent revealing innate preferences towards this mode, but would verify if confidence in waypoint control improves performance. Pilots should also have been warned about making waypoints that are too far apart, as the drone will pick up speed and naturally drift slightly from a straight line path, sometimes drifting past the desired waypoint as it slows down from such a speed. This is possibly a problem with the system being used and its failure to exactly replicate what the operator desires, but this seems much more realistic than the alternative of capping acceleration at a very low value and having a very large window to begin slowing the drone as it approaches each waypoint. This accounts for much of the concerns that participants voiced in the post experiment survey about the interface not being accurate. While it may be desirable to keep it as such, training operators on this seems to be a real world nuance that fits well with the goal of SC+.

These are important elements to add in future trainings, but the failure to add them even on the second iteration demonstrates how difficult it sometimes is for people familiar with the system to recognize the behaviors that operators are most likely to need to be discouraged from adopting. Even pilot testing can have trouble catching this, as one-time participants are not likely to notice overall behavioral trends, even if they exhibited them.

Looking at the crash maps from these experiments can help suggest some of the common problems to bring up in these additional trainings. Figures 4.1 and 4.2 show an overview of where crashes took place on the map for Experiments 1 and 2, respectively. Note that the bolded arrows with notched tails in Figure 4.2 indicate crashes that took place on the return trip, while X's indicate a forced touchdown (either out of battery or, in one case, a participant accidentally landing inside a barrel). The tunnel was far and away the largest crash point for participants in both experiments. This is not unexpected, as the entrance to the tunnel is somewhat narrow, and the tunnel exposed weaknesses in operator strategies by either forcing inexperienced users to interact more with inspection mode or causing unobservant participants to incorrectly line up waypoints and crash into the tunnel entrance or exit. Walls, corners, barrels, and entrances also proved to be common crash points, as they forced users to employ more creative navigation techniques.

This partially matches with the results from the post-experiment feedback from the second experiment: there, participants cited the tunnel as the most challenging part to navigate through, with the wall in the second half of the course and the barrels also posing a challenge for individuals. However, while the barrels represented the second largest proportion of the crashes, it seems that participants found navigation around the wall appeared much more challenging. While this could partially be because many participants did not succeed and never made it back to deal with the barrel obstacles a second time, it also fits well with the aforementioned lack of spatial understanding; the wide-angle lens likely made the space by the wall seem very narrow to unfamiliar participants. Providing a clear indication of this spatial awareness both in the training and in the app interface can likely improve performance and may change overall user strategies.

4.2 Designing an Effective System

One of the most striking things to note about these experiments is that almost every piece of feedback from participants called out system limitations rather than problems they had with the trainings. Teaching spatial awareness and perception may be one exception, but this was an unintentional problem of a wide-angle camera designed for outdoor use and a low-quality video feed presented in-app rather than a deliberate attempt to watch how strategies changed to adapt to this limitation. As it turns out, building a system to run these types of experiments is not at all trivial, not least because a single issue in the relatively short span of an experiment can dramatically alter how participants feel about the tools that they have been given. Improving this system design not only makes these experiments more scalable and

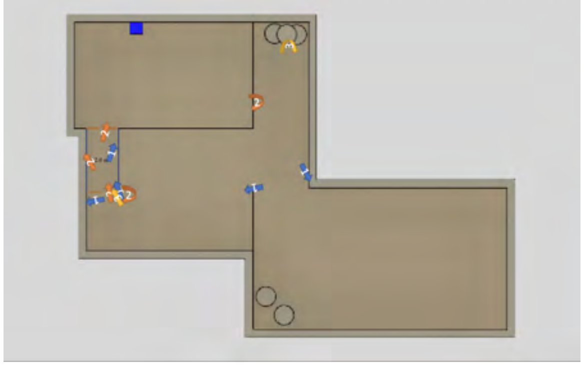
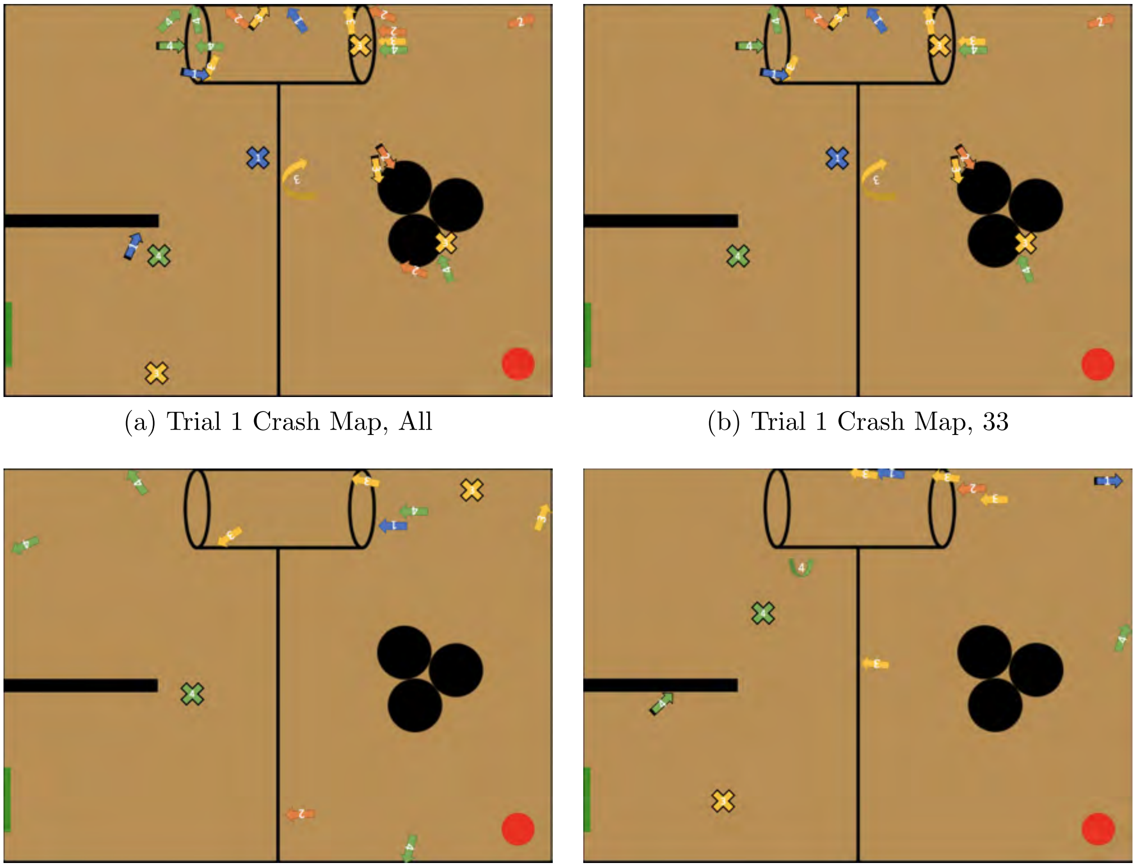


FIGURE 4.1: Experiment 1 Crash Map



(a) Trial 1 Crash Map, All

(b) Trial 1 Crash Map, 33

(c) Trial 2 Crash Map, 33

(d) Trial 3 Crash Map, 33

Figure 4.2: Experiment 2 Crash Maps

easier to maintain for the experimenters involved, but it also reduces the chance that operator strategies are adopted because of system factors instead of the trainings being tested. While it is difficult to measure to what extent system limitations influenced the outcomes of these experiments, it is at least likely that some of the high degree of variance in completion times came from non-optimal strategies caused by these system limitations.

4.2.1 System Design

One major change that came about between the experiments was the dramatic reduction in size of the room. This was largely a response to the complexity of maintaining a motion capture environment in such a large and complex space. However, two observations can be made in considering the ideal future environment for these experiments:

1. Many participants are failing the experiment.

2. There are some partial or completely invalid results due to system error.

The first observation does not necessarily have to be fixed–adding more robust data collection also increases the range of conclusions that can be drawn, even from participants who fail to complete the course. However, intelligent system design on this issue is crucial to collecting a complete picture of operator engagement. On one hand, if participants fail to complete the course in full, then they are never exposed to parts of the course that data could have been collected from. On the other hand, a course that is too simple makes it harder to draw distinctions between groups and produces fewer unique operator strategies to analyze.

The second observation is also the result of several factors and is never completely avoidable, but it is unequivocally true that a larger space is more difficult to comprehensively cover with a motion capture setup. Each additional obstacle and wall can substantially reduce the visibility of previously placed cameras, requiring very deliberate camera placement to cover a large space with moving obstacles. It's not enough to have good general coverage. If the drone is lost for a second while moving, then it will likely be rediscovered fairly quickly in its path; however, if the drone is lost while hovering in the same spot, it will be unlikely to recover its position before a touchdown is triggered. In this relatively simple dichotomy, it may be possible to program a logical flow dictating what the flight controller should do when it loses sight of the drone based off previous position, speed, etc. At large scales, however, experimenters will eventually need to begin purchasing more cameras to get coverage that is "good enough." Even with good coverage, one common issue is that repeated crashes of the drone will damage the reflective paint on the markers that motion capture systems use to track objects. In this case, redundant coverage of areas is crucial to ensure that vision isn't lost because of an angle of the drone that is harder for the camera to pick up. With all this in mind, there is a great deal of value to experiment maintainability in using a smaller space.

As a complication, there are some inadvertent outcomes of a more confined course that may have contributed to the highly variant completion times from these experiments. First, the shorter course may have actually changed strategies by incentivizing the use of inspection mode in some places where it should not be used. In larger areas, placing a waypoint to quickly traverse a room that is known to be empty may be a much more reasonable strategy; in smaller rooms, simply moving the drone slightly using the thumbpads may appear to make more sense. The overall amount of inspection mode usage might remain the same, as it is still likely to be relied upon when arriving at thresholds such as the entrance to a new room or to the tunnel, but increased usage of waypoint control might have convinced more participants to rely on this autonomy more frequently. Good training that compares the two modes and their strengths/weaknesses will likely play into how much this effect occurs. Second, a shorter course increases the amount of time the drone spends in low coverage regions. As previously mentioned, the drone is much more likely to crash upon being lost from motion capture vision if it is hovering in the same area. With a smaller course, there is a corresponding increase in the chance that the drone is hovering in a region with problematic coverage. This isn't an argument against smaller, more well-designed courses, but it reiterates the importance of establishing good coverage. Finally, smaller courses expose the operator to more time with the drone near pain points like walls where kickback destabilizes the drone. Effective trust in semi-autonomous control relies on the assumption that the system will hover in place effectively. If it appears that this is not the case even in a single instance, operators could see good reason to try and move too quickly or to adapt to increased use of inspection mode to manually correct for sway and instability, even when their manual inputs may actually be worse in the long run. Operators who saw a crash from this hover sway during the module trainings might account for some of the outlier cases where participants flew the entire course in inspection mode.

4.2.2 Interface Design

In many ways, the app interfaces used for these experiments were designed to be as simple as possible. Neither of these experiments was trying to see how particular interface choices improved ease of use for drone operators, though this is a plausible and interesting area to pursue for further work. Making these apps as bare and simple as possible helps to preserve a focus on the training itself. However, the interface is what is truly being critiqued when participants addressed concerns about lag, stability, reliability, or spatial awareness. This is fairly reasonable: a participant with almost no experience using an app has a limited number of reference points to judge optimal strategies, so even a single perceived app failure is likely to strongly color their choices in the direction of whatever they were more comfortable with before starting the experiment. In both experiments, some participants seemed to strongly favor using inspection mode for substantial amounts of the course, despite being told that waypoint control was faster and that the goal of the course was to have the fastest time possible. This might have been a result of making either inspection mode too familiar to use or failing to make the supervisory interface intuitive enough, but it is certainly true that addressing some of the most common interface problems will help increase understanding of the autonomous interface.

Of the several interface issues that may have decreased trust in parts of the supervisory control, the camera feed is the most worthy of addressing. Eliminating the latency and improving the quality in the camera feed was a major problem during app development; while it was substantially improved, it can still reach several hundred milliseconds in latency with low quality levels that do not pair well with a wide angle lens designed for outdoor use. Though some latency is certainly a part of any real-world system, the low quality makes this level of latency a bit too high compared to a viable commercial application. Several participants mentioned this in person to experimenters and in the post-experiment survey as forcing them to change their strategy to make very slight movements, wait to see how the drone reacted, and then respond accordingly. Participants who either failed to perform well in inspection mode or relied too much on the waypoint interface due to distrust in the inspection mode interface would likely benefit substantially from this change, and it would be interesting to see if this boosted performance for operators exposed to both MC and SC.

On a similar note, several participants also suggested that the main supervisory interface feature the camera feed in the bottom left or top left corner so that participants do not need to stop after every waypoint, pause the flight plan, and then inspect the room before continuing. This seems like a fairly useful feature to add, particularly in a smaller, more tightly packed course, where it is difficult to make more than small movements without needing to check surroundings. Indeed, it seems reasonable that this particular feedback may not have been given for the first experiment because a larger course allowed the drone to fly to a start of the room, inspect the area (which usually had larger paths to navigate through), and then fly across it. This kind of change would reduce the overhead of mode-switching and help eliminate mistakes caused purely by the challenges of the interface.

4.3 Further Work

Future analysis on this experiment will follow the path of the work published from J. Zhou of using the existing data set and analyzing general strategies and more specific time breakdowns using Hidden Markov Modeling. [14] In addition to providing more understanding about how operator workload and thinking are occurring, this will help identify the actual proportion of time spent in inspection mode to further evaluate the qualitative claims made in this piece about observed trends for these modes of control. The knowledge established here with regard to experimental and interface design can also help future researchers eliminate noise to collect more robust data. In particular, it would be interesting to generalize the strategies used by operators and see how those change when the interfaces themselves vary–some interesting additions might be to see how inclusions of higher, more protective forms of autonomy like oncoming collision detection change these results.

5
Conclusion

As semi-autonomous systems continue to proliferate in traditionally manual or human applications, understanding how to optimize human efficiency within these systems will remain a central challenge. UAVs in particular offer an excellent opportunity to give humans the power to reach difficult to access areas or mechanize processes that require large amounts of movement.

The system designed in these experiments attempted to create a simulated environment to help understand these applications and their associated challenges. Both experiments using this system demonstrated that additional training frequently leads to unexpected consequences by unintentionally encouraging participants to adopt certain strategies. In particular, additional training in some environments can sometimes encourage overreliance on the supervisory control because of its comparative ease of use over a lower autonomy control scheme. Conversely, excess supervisory training in smaller systems may sometimes encourage the opposite effect, leading participants to spend too much time in inspection mode when the tradeoff of one system versus the other is less clear. In general, more nuanced training that focuses on integration between modes is essential to producing high-performing operators.

At a larger scale, the challenges in deploying a system like this suggest how important system design is in bridging the gap between humans and machines. Technical challenges like user interface design, camera latency, and relative stability seem like questions of overall data reliability, but are actually likely to influence operator behavior in cases where participants have limited interaction with the system. Motion capture environments offer an excellent opportunity to manufacture simulated environments for semi-autonomous systems and provide a substantial amount of data for experimenters to collect, but they still require a substantial amount of work to ensure that the corresponding testing is not biased. Incorporating understanding about mode confusion and operator workload into these design choices will allow for more robust and less noisy data, while understanding preconceptions and problems with these interfaces and environments can also inform a better understanding of operator strategy modeling.

Bibliography

[1] Yves Boussemart et al. “Supervised vs Unsupervised Learning for Operator State Modeling in Unmanned Vehicle Settings”. In: *Journal of Aerospace Computing, Information, and Communication* 8 (2011). DOI: 10.2514/1.46767.

[2] M. L. Cummings et al. “Automation architecture for single operator, multiple UAV command and control”. In: *The International C2 Journal* 1.2 (2007), pp. 1–24.

[3] James T. Hing, Keith W. Sevcik, and Paul Y. Oh. “Improving unmanned aerial vehicle pilot training and operation for flying in cluttered environments”. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2009). DOI: 10.1109/IROS.2009.5354080.

[4] Minwoo Kim. “The impact of skill-based training across different levels of autonomy for drone inspection tasks”. Master’s thesis. Duke University, Durham, NC. 2018.

[5] Ahmed Mashood et al. “A hardware setup for formation flight of UAVs using motion tracking system”. In: *2015 10th International Symposium on Mechatronics and its Applications (ISMA)* (2016). DOI: 10.1109/ISMA.2015.7373474.

[6] Francesco Nex and Fabio Remondino. “UAV for 3D mapping applications: a review”. In: *Applied Geomatics* 6.1 (2013), pp. 1–15.

[7] Nikhil Nigam et al. “Control of Multiple UAVs for Persistent Surveillance: Algorithm and Flight Test Results”. In: *IEEE Transactions on Control Systems Technology* 20.5 (2011). DOI: 10.1109/TCST.2011.2167331.

[8] *Paparazzi UAV*. https://github.com/paparazzi/paparazzi. 2003–.

[9] David Pitman and M. L. Cummings. “Collaborative Exploration with a Micro Aerial Vehicle: A Novel Interaction Method for Controlling a MAV with a Hand-Held Device”. In: *Advances in Human-Computer Interaction* 2012 (2012). DOI: http://dx.doi.org/10.1155/2012/768180.

[10] *PPRZonDroid*. https://github.com/paparazzi/PPRZonDroid. 2014–.

[11] Victor Rodriguez Fernandez, Antonio Gonzalez-Pardo, and David Camacho. “Modeling the Behavior of Unskilled Users in a Multi-UAV Simulation Environment”. In: Oct. 2015. DOI: 10.1007/978-3-319-24834-9.

[12] Gabriel G. De la Torre, Miguel A. Ramallo, and Elizabeth Cervantes. "Workload perception in drone flight training simulators". In: *Computers in Human Behavior* 64 (2016), pp. 449–454. DOI: https://doi.org/10.1016/j.chb.2016.07.040.

[13] Jie-Tong Zhou, Chi-Yi Wang, and Yueh Min Wang. "The development of indoor positioning aerial robot based on motion capture system". In: *2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE)* (2017). DOI: 10.1109/ICAMSE.2016.7840347.

[14] Jin Zhou et al. "The Impact of Different Levels of Autonomy and Training on Operators' Drone Control Strategies". In: *ACM Trans. Hum.-Robot Interact* 1.1 (August 2018). DOI: https://doi.org/10.1145/nnnnnnn.nnnnnnn.