University of Nevada, Reno

A Realistic Simulation for Swarm UAVs and Performance Metrics for Operator User Interfaces

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Computer Science and Engineering

by
Kazi Tanvir Ahmed Siddiqui

Dr. David Feil-Seifer/Thesis Advisor

August 2017

THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

KAZI TANVIR AHMED SIDDIQUI

Entitled

A Realistic Simulation for Swarm UAVs and Performance Metrics for Operator User Interfaces

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

David Feil-Seifer, Advisor

Sushil Louis, Committee Member

Hao Xu, Graduate School Representative

David W. Zeh, Ph.D., Dean, Graduate School

August, 2017

Abstract

by Kazi Tanvir Ahmed Siddiqui

Robots have been utilized to support disaster mitigation missions through exploration of areas that are either unreachable or hazardous for human rescuers [1]. The great potential for robotics in disaster mitigation has been recognized by the research community and during the last decade, a lot of research has been focused on developing robotic systems for this purpose. In this thesis, we present a description of the usage and classification of UAVs and performance metrics that affect controlling of UAVs. We also present new contributions to the UAV simulator developed by ECSL and RRL: the integration of flight dynamics of Hummingbird quadcopter, and distance optimization using a Genetic algorithm.

Acknowledgements

First, I would thank the Almighty for giving me determination and patience to write this thesis. I would like to give my heartfelt thanks to my advisor Dr. David Feil-Seifer, who contribution to this work is unimaginable.

I thank the other members of my defense committee Dr. Sushil Louis and Dr. Hao Xu. Without their generous help, guidance, and motivation this thesis would be incomplete.

I would also like to thank Rumit Kumar and Dr. Manish Kumar from University of Cincinnati, without their generous help the Flight Dynamics chapter would be incomplete. I am also grateful to Dritan Hasani for his content about the user interface of UAVs.

Finally, I would like to dedicate all my hard work to my mother, without her sacrifice and constant support I would never come this far.

Contents

List of Figures

Chapter 1

Introduction

Unmanned aerial vehicles (UAVs) have been utilized for both military and civilian applications. The successful deployment of drones in military missions encouraged many other countries and governments to start investing in drone development programs, hence modern unmanned aerial vehicles hold an important and permanent position in the military arsenal of the US and many other countries across Europe, Middle East, and Asia. In addition to the military usage, there is also a great potential for using UAVs in numerous civilian applications. In recent years the research and business communities became highly interested in exploring the possibilities of using UAVs for civilian applications such as fire fighting, search and rescue operation, product delivery, surveillance, construction and building inspection, film and television production, mapping, agriculture, etc. However, the methods for controlling these UAVs can vary greatly due to the nature of the tasks assigned to the UAVs, the communication possible between UAV and operator, and the regulatory environment the UAVs are operating within. In this thesis, we will examine UAV user interfaces, for a wide range of scenarios. We also introduced the simulation of an autonomous Hummingbird quadcopter flying through user specified points in Reno, Nevada. We used genetic algorithm [3] to optimize the total distance travelled by the UAV.

Despite the current uses for UAVs, there are numerous challenges that need to be addressed before their full potential can be utilized in the daily civilian applications. A number of technical challenges related to navigation, sensing, communication, bandwidth, autonomy, etc. need to be addressed before UAVs can be integrated into the civilian airspace. Because of these concerns the US FAA has in place a set of regulations that make it illegal to fly UAVs into the regular airspace or to conduct flights over densely populated areas. Williams [4] suggests that the rate of accidents for UAVs is several times higher than for manned aircraft. The development of UAV user interfaces (UI) becomes more important as the FAA considers more autonomous and beyond line-of-sight (BLOS) operation could soon be possible under proposed regulation.

Crucial factor for UAV is the user interface design. For using UAVs in search and rescue operation, which is often time critical and demands lots of attention from the operator, the design of the user interface demands much attention. Numerous research showed that the design of the user interface affects the awareness, workload and performance of the operator.

The remainder of the paper is organized as follows:

- **Related Work**: We describe significant research based on their ingenuity and clarity for UAV classification and User Interface design. We also provided a historical overview in this section so that a reader can understand how the design evolved overtime.

- **Level of Autonomy**: In this chapter, we discuss level of autonomy, which is one of the most important and contentious part in controlling UAVs. We highlight research that defined and classified different levels of autonomy. We also review how autonomy level affects interaction with UAVs, workload, and performance.

- **Performance Metrics for Sensory Control**: This chapter is a collection of literature on different performance measurement of UAV interfaces. We reviewed many performance metrics. We included as many performance metrics as possible from various sources, and tried to combine them in broad categories.

- **UAV Simulator**: We describe the simulator architecture for UAV operators in Reno, Nevada. We introduce a disaster zone that was built in Unity game engine to simulate semi-autonomous flying of a Hummingbird quadcopter. We also describe a genetic algorithm [3] for optimizing the total distance travelled by the UAV.

- **Measuring Awareness and Workload**: This chapter describes an experiment we set up to show the correlation between level of autonomy versus situational awareness and operator workload. We used the NASA-TLX scale in this experiment to show operator training effects.
- **Conclusion and Future Work**: We summarize the contributions of this thesis, and address critical problems that need to be solved in future.

1.1 Summary

UAVs are used for of civilian purposes such for: recreational photography, farming, surveillance, drone delivery, and search and rescue operation. The increasing usage of UAVs elements that we think about the user interface design for controlling the UAVs, and their effects on operators' performance. An increasing tendency towards autonomous systems also instigated us to develop the autonomous path planning and distance optimization by UAVs.

Chapter 2

Related Work

Early systems used *teleoperation* as the main method for controlling UAVs. In this approach a human operator sits at a ground station and maintains UAV control from a distance. The control interface could be a joystick, waypoint navigation through a graphical user interface, virtual reality headset, or any other innovative interface, but the connotation of teleoperation is that the distance is too great for the operator to see what the UAV is doing therefore the interface must have some type of display and control mechanisms [5]. The major drawback of teleoperation is that it requires at least one human operator per UAV, possibly more depending on mission objectives. For example, some military UAVs require up to four human teleoperators plus a fifth who specializes in takeoffs and landings [5]. To eliminate these drawbacks *semi-autonomous control* was pursued, where the UAV and human operator share control over the system. In general, a UAV has a set of lower-level operations that it can perform and an operator issues high-level commands that will then be executed in a closed loop fashion by the UAV. Previous research has shown that a semi-autonomous approach is well accepted by the emergency response community since they have more confidence in a system that allows for such control [6, 7]. This thesis will refer to the semi-autonomous approach for UAV control.

For both *teleoperation* and *semi-autonomous control*, the operator controls the UAV from a distance, therefore it is necessary to have some type of interface not only to issue commands but also to see what the UAV is doing. Previous work has addressed the issue of achieving multi-robot control through well-designed interfaces that take into account the cognitive and perceptual strengths and limitations of the human operators [8, 9]. Others have focused on studying the user requirements and their implications for the interface design, showing that interfaces are often overloaded with unnecessary information, potentially causing data to be neglected by operators [10, 11]. UAV literature generally discusses the human factors in terms of *situational awareness* (SA) and *operator cognitive load* (CL). When discussing the human factors associated with the design of interfaces for multiple UAV control our main focus will be on the human—UAV interaction in terms of CL and SA. We are interested in exploring if human factors have been adequately addressed in the current UAV interface development, and subsequently identify the best practices for UAV interface design, and establish the appropriate levels of autonomy in order to optimize CL and SA. For example, designers wish to maximize operator performance while simultaneously reducing the workload of the human operator.

Research has shown that the content and format of information displayed in the interface has a potentially large effect on the system's performance level [12]. Information that is well organized, provides enough concrete details, and is consistent tends to increase the level of trust that the operator has for the system. Baker *et al.* (2004) have shown the importance of integrating visual information from incoming video with the other robot sensor information [11]. Their study revealed that during a disaster mitigation mission most of the users, except for those that were highly experienced, focused solely on the video-stream window, while neglecting other information on the interface screen. On the other hand, simply incorporating all the necessary information around the video window is not a solution since it may overload the display and therefore increase the workload on a human operator. To overcome the issue of increased monitoring requirements it is desirable to "hide" certain information from the user and utilize "alarm-systems" to notify the operators as needed about the critical events that need their attention. This leads to another research question related to what type of information can be "hidden", and at what mission stages can this information be "hidden" without having any negative impact on the system performance in general, as well as on the operator's situational awareness in specific?

Another human factor to be considered is related to the question of what sensory input should be used to communicate important information to the operator, visual (pop-up windows), auditory, or a combination of both? Audio alerts introduce less overload on the operator, but in the other hand they provide less situational awareness as the operator needs to distinguish what a given signal is telling them. Wickens (2002) suggests that cross-modal approach (information divided between one visual and one auditory channel) is better than intra-modal (e.g. two visual channels) [13]. Finding the trade-off between these two factors is an interesting topic that needs to be further investigated through experimentation. Further work needs to be done in order to determine precisely what type of information is best presented visually and what through audio signals, and how to integrate the two sensory inputs to achieve an improved situational awareness while reducing the operator workload.

2.1 The Case For Simulation

A review of mobile robot simulation environments reveals that simulation is becoming an increasingly important aspect of mobile robots [14], helping researchers perform more experimentation in this area. A realistic graphical rendering system and ideal physics simulations are the main features of a good simulator. Computer video game engines often are used to power a robot simulation environment capable of simulating multiple robots, people, and objects in the environment. Most game engines were built for creating physically accurate simulation in the game world. We utilized this physics engine to model the world where our UAV will be operating. This simulation might not be perfect in terms of the physics governing the universe, but a close approximation of how a real world UAV would behave.

We simulated the flight dynamics of a Hummingbird quadcopter. For the autonomous flying of this UAV, Michael et. al. [15] developed a mathematical model. To move the UAV from point A to point B, the error in x, y, z position are calculated and used to determine δ_{a_x}, δ_{a_y}, and δ_{a_z}. To calculate the angular speed for each of the rotors and their orientation, the changes in rotation angles e_θ, e_ϕ, and e_ψ are calculated. Using these values, the angular speed of the motors: Ω_1, Ω_2, Ω_3, and Ω_4 can be determined. Using the values of angular speed of the rotors, we calculate the propeller force, moments, and inputs for the rotors. Thus, the UAV advances to its destination.

2.2 Summary

In this chapter we reviewed the literature related to UAV user interface. The review shows that there have been numerous work done in the field of UAV. We given a brief background on how UAVs evolved overtime, and what are the possible ways to improve them in future. We also discussed the importance of simulation in disaster mission using UAVs.

Chapter 3

Level of Autonomy

Autonomy of a UAV refers to the independence it has for navigation, coordination, and decision making. A fully autonomous UAV should be able to accomplish its mission without any human intervention. For example, in a search and rescue mission, the UAV would need to identify a target, plan the shortest path towards that target, and deliver the payload in a fixed time, while maintaining communication with other UAVs in the fleet, and properly selecting tasks based on priority. It should also have a safety mechanism so that a mid-air incident does not occur and cause harm to anyone. Manually controlled UAVs are controlled by one or more operators who maintain the speed, altitude, flight-path, manage any payload, check the fuel and other status, avoid any obstacles, and make decisions about the target. Current UAV usage primarily utilizes manual control, which causes heavy workload and less awareness to the operators.

The level of autonomy plays an important role in the performance of UAV operators. According to Durst and Gray [16], three of the biggest challenges for autonomous vehicles are: user acceptance of the unmanned system, effective test and evaluation of autonomy, and defining a universal autonomy level for the system. This relates directly to the trust issue discussed above (and expanded further below). Though there is novel research on integrating autonomy in UAVs, standards for UAV autonomy remain elusive. The authors [16] categorized the frameworks for universal autonomy level into two categories: contextual and non-contextual. Contextual methodologies take account of the UAV's mission complexity, environmental complexity, and human independence. Mission complexity includes commanding structure, type of tasks, collaboration, planning etc. Environmental complexity includes terrain structure, object's density/types, weather condition, threats decoy, and mapping etc. Human independence includes interaction time, planning time, interaction level, workload etc. Non-contextual methodologies do not take account of these factors.

Automation also influences the level of workload on human operators. While intuitively we might think that higher automation by default means less overload, this is not always the case. Automation can often have the effect of merely changing the nature of workload (reduce manual load while increasing the cognitive load), or shifting the workload in time (support the pilot at times of low workload but fail to do so when needed most). The research has shown that increased automation might in occasions have such negative effect, thus the UAV system should be designed not only to avoid overload, but under-load as well [17].

The level of autonomy may have bearing on the trust of UAVs in search and rescue. Automation would help to reduce operator error and cognitive overload, but on the other hand, previous research has shown that the emergency personnel have little trust in fully-autonomous systems, since fully-automated systems need to be governed by formal processes that would limit their flexibility to address certain situations. Therefore, tele-operated, semi-autonomous, rather than fully-autonomous UAVs would be preferred. It might be advantageous to have a human operator ready to take over when automation reaches its limits [10].

The prior paragraph suggests that reducing operator error through autonomy while increasing operator trust in a UAV system may affect adoption of such technology. Riley [18] showed that trust is the main factor that determines if an operator will choose to use an optional automation or refuse to accept its usefulness altogether. Research has shown that both extremes of trust can be potentially dangerous for the success of the mission. Insufficient trust can lead to situations where the operator refuses to make use of automation [19], while in the other hand over-trust may lead the operator to rely on automation even in situations when better performance would be achieved if the human took control over the system [20]. These issues need to be taken into account when designing the human-UAV interaction strategies, and this leads to an important research question: how to establish the appropriate balance between autonomy and human control in order to achieve maximum trust and acceptance for the UAV [21]?

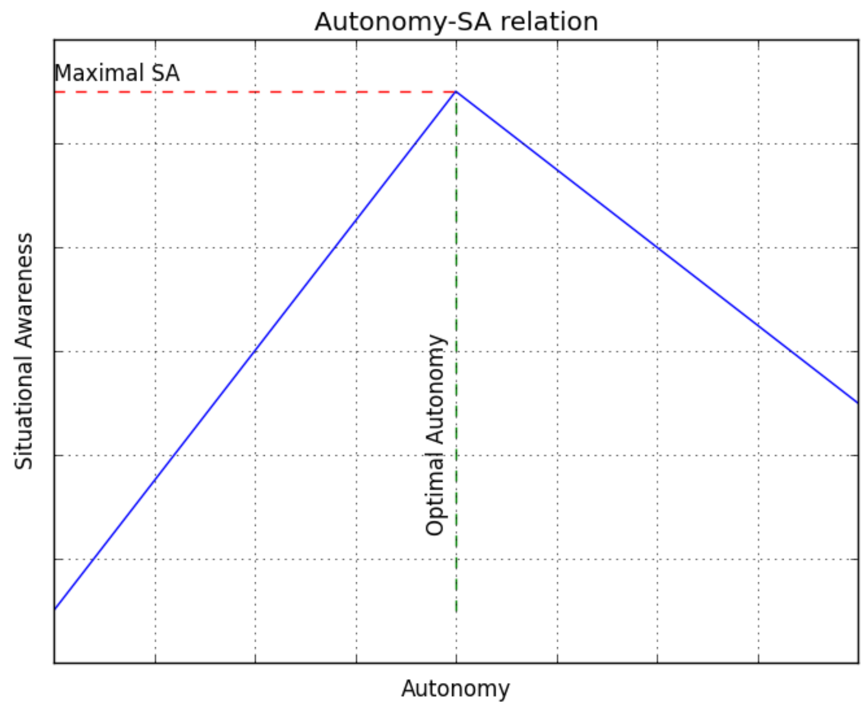


FIGURE 3.1: Autonomy vs. Situational Awareness. Situational Awareness rises when autonomy rises upto a fixed level. After a threshold point, it decreases.

Another implication of semi- or full-autonomy is its effect on operator alertness or situational awareness during the disaster mitigation mission. Bainbridge, et al., have shown that operators have difficulty maintaining their attention during periods of low task demand [22]. Hence, operator under-load should be taken into account because reduced situational awareness may resultand result in the failure of a human operator to observe important (even critical) moments during a disaster mitigation mission. The effects of automation on SA are best described through the graph in Figure 3.1. From the figure we can see that the SA increases autonomy up until the point when in reaches the maximal SA. Increasing the autonomy beyond this point has the negative effect of reduced SA because of the low task demand. Further research should be conducted in order to ascertain the *optimal autonomy* level that ensures the maximal SA in a disaster mitigation scenario.

3.1 Summary

Research in autonomy level shows that for a disaster mitigation robot, it is better to use a semi-autonomous robot rather than a fully autonomous robot. Semi-autonomous robots give an operator the ability to share control to maximize joint efficiency. In that way, the principle control of the system retains in the human hand. Other research showed that people tend to rely less on fully autonomous robots. The level of autonomy deserves a lot of discussion, as it is related to trust, awareness, and workload. We presented an experiment in later chapter, where we tried to find a correlation among level of autonomy versus situational awareness and workload.

Chapter 4

Performance Metrics For Sensory Control

Our goal is to design an interface that produces optimum level of cognitive load and awareness. For this reason, it is important to measure some factors and relate them with the performance. Murphy emphasized the importance of designing common metrics that would become the standard for evaluating HRI aspects in the robot-assisted SAR systems [23]. However no such standard has been universally accepted and researchers use different metrics that are often contradictory, evaluate interfaces incorrectly, or evaluate one aspect of human-robot interaction correctly but fail to properly validate the system as a whole. In this chapter, we discussed some performance metrics which are related to the user interface for controlling multiple

UAVs. We included mentioned numerous articles and tried to segment all the factors responsible for performance metrics in broad categories.

Several researchers have taken the approach of measuring the system performance, hence workload and situational awareness, in terms of time required for an operator to learn using the interface, time to complete a given mission, and minimization of critical incidents during the operation [24]. In an attempt to compare operator performance in supervisory vs. manual control, Geddes et al.[25] assessed the controller task demand by measuring the number of actions required to complete the task, cognitive workload, and time required for the task. The operator's tasks were divided into three hierarchical stages. In the first case, the operator controlled the aircraft directly using commands for pitch, roll, thrust etc. In the second case, the operator set the course, altitude, and airspeed. In the third step, the operator issued task level commands such as *line formation* and *trail formation*. The experiment showed that, in terms of task performance and cognitive workload, supervisory control required fewer number of actions by the operator and much less cognitive workload.

Operator workload, trust in automation, operator multitasking performance, and situational awareness are among the most critical factors for assessing performance of UAV control. However, there are many other factors which may have significant effects on the performance of UAV operator using a particular control interface design. We decided to divide these factors into two broad categories: Subjective and Objective measures.

Subjective measures are the self-measurement of operator performance. These can include: perceived difficulty level, physical and mental workload, effect of provision, trust in automation, stress, anxiety, frustration etc. Objective measures are explicit assessments of user performance, such as situational awareness, operator multitasking performance, number of targets detected, area of the map covered, number of UAVs controlled simultaneously etc.—basically the data which correspond directly to the goal of a specific task. Some factors can be regarded as both subjective or objective, and can be measured in both ways. The next two sections will explore these measures in more details.

4.1 Subjective Measures

The subjective measurement scale evaluates the system performance from an operator's perspective by measuring the amount of information retained in working memory [26]. Subjective scales are more practical [27] and also the easiest method to assess workload [26]. Measurements are taken during task or after the task. Casner and Gore [28] categorized the subjective measurement scales in two groups. One asks the operator to assign a numerical value for a particular task, while the other asks an operator to compare the tasks according to difficulty and/or workload. Most of these subjective measures use an n-point Likert scale. Some examples are: Instantaneous Self Assessment, Bedford, NASA Task Load Index (NASA-TLX), Subjective Work

Assessment Technique (SWAT), Modified Cooper-Harper scale, Dynamic Workload scale, Overall Workload scale [27] etc.

John et. al. [29] also showed types of subjective rating scales and their strengths and weaknesses:

Rating Scale	**Rating Scale Type**	**Strengths/Weaknesses**
Air Force Flight Test Center (AFFTC)	System Adequacy, 6 point., Interval, Bi-Polar	Strengths: No middle point-forced to make choice. Weaknesses: Not used outside AFFTC
AFFTC-modified USAF-SAM (School of Aerospace Medicine)	Workload, 7 point, Interval	Strengths: Easy to use, fits on flight cards. Weaknesses: General workload not specific
Readability and Strength	Comm Quality, 5 point., Interval	Strengths: Pilot friendly/familiar. Weaknesses: Verbal anchors not defined
Bedford	Workload, Ordinal	Strengths: Pilot friendly/familiar, validated [30]. Weaknesses: Mid-range semantic descriptors are vague, interchangeable
Subjective Workload Assessment Technique (SWAT)	Workload, 100 point, Interval, 3 Dimensional	Strengths: Easy to use, once learned. Weaknesses: Requires card sort, specialized software, and training
Modified Cooper-Harper	Workload, Ordinal	Strengths: Pilot friendly/familiar. Weaknesses: Non-interval scale
Situation Awareness Global Assessment Technique (SAGAT)	Situation Awareness	Strengths: Widely used and accepted. Weaknesses: Limited to simulations.

Table 4.1: Subjective Human-System Integration Measures of Performance

4.1.1 Operator Workload

Operator workload can be defined as the level of work or attention required from the operator in order to complete the mission in a successful and efficient manner. Workload can be both physical and mental. In the context of operating UAVs, we will only consider mental workload. Present works of measuring workload focus on psychomotor, perceptual, or communication workload [31]. Nisser and Westin [32] defined workload as the total amount of demands put on an operator and the subjective response of that operator to those demands. Gopher [33] described workload as a cognitive resource required to perform a task. We believe that, operator workload for controlling single or multiple UAVs, is a temporary state of mind which indicates the amount of concentration required for successfully accomplishing a task. It can be measured by monitoring brain activity while in the task. We also hypothesize that more workload does not ensure better performance, rather we believe that performance can be enhanced in a supervisory control task by becoming more familiar with the system and interface.

Many different methodologies to measure workload exist. Miller [27] categorized the workload measurement into three groups: physiological, subjective, and performance-based measures. In physiological measures, the operators heart-rate, eye blink rate, brain activity, blood pressure, respiratory rate etc. are measured. The underlying belief is that sudden physiological changes indicate workload. Subjective measure asks the operator to rate the overall task. There are numerous methods for subjective measures, as we discussed earlier. Finally, the performance-based measurement tries to estimate the workload from operator's difference in performance versus the difference in workload. Casner and Gore [28] also has the same classification, except that they added a new category—Indirect Measurement. In Indirect Measurement, workload is measured by adding a secondary task along with the primary task, to measure how much *spare capacity* the user has. They also proposed to measure speed, accuracy, and activity during the task as an indication of physiological workload.

4.1.2 Trust in Automation

There are three components of trust which are found almost universally. First, there must a *truster* who put his/her trust on someone; there must be a *trustee* who was trusted, and something is at stake in this relationship. Second, the trustee must have an incentive, for example money, goodwill, or reputation to hold the trust. Finally, there must a possibility that the trustee may break the trust [34]. According to Lee and See [21], trust is the expectation of a favorable outcome from opposite end. Trust is an untenable component in interpersonal relation. It is also very significant in human-machine interaction [35].

Autonomy enables machines to follow particular patterns repeatedly without any errors. It is the technology that "actively selects data, transforms information, makes decisions, or controls processes" [21]. Human-automation labor systems can be very efficient and give people more freedom [35]. Studies showed that trust in automation has both beneficial and deleterious effects. In a study conducted by Dzindolet et. al. [36], people were asked to identify the presence or absence of a camouflaged soldier in slides of a terrain using a decision making aide; most of their decisions were biased by the aide. However, when they found out that the aide made mistakes, they hesitated to utilize the aides, even if trust had been established prior. Chen [37] pointed out that trust in automation is misleading as it has the "connotation of a prescribed behavior." *Calibration* is a more appropriate term in automation. Because, operators will only intervene in the supervisory task when they believe that their decisions are superior than machines'.

The level of reliability is also crucial to trust. Wickens and Dixon [38] showed that, "a reliability of 0.7 was the 'crossover point', below which unreliable automation was worse than no automation at all." When there is a lack of reliability on autonomous system than a manual one, people tend to misuse that system. Mosier and Skitka [39] hypothesized that people rely on autonomous systems because they believe that these systems are more reliable than manual ones. In addition, reliable automation significantly reduces decision time compared to manual performance [40]. Therefore, in terms of decision making and reliability, trust in automation is very important. As UAV operators have to deal with autonomy and decision making continuously, trust is a major concern for building a reliable user interface for UAVs.

4.2 Objective Measures

In the previous section, we described how subjective measures contribute in the performance of UAV operators. Though subjective measures are very important for assessing performance, objective measures are the direct indication of system performance [29]. For example in a human-robot cooperation task, objective measures would measure the robot tasking time, mission execution time, and switching time [41]. De Visser et. al. [42] measured robot performance and team performance as an objective measure in another human-robot collaboration task. They defined *Execution Efficiency* and *Navigational Efficiency* as two metrics to measure robot performance while team performance was measured in MITPAS. Execution Efficiency is the ratio of the time for executing a task vs. the total mission time. Navigational Efficiency is the actual distance travelled by a robot compared to the preplanned route length.

Other researchers came up with different methods to implicitly measure operator performance. Those include measuring physiological indication as mentioned earlier, anthropomorphic measures which evaluates pilot's surroundings in the cockpit, perceptual measures evaluates the quality of display or auditory feedback system, and operator/system performance is measured by flight control quality, situational awareness, and workload measurement [29]. Situation Awareness Global Assessment Test (SAGAT) is a widely used technique to assess the situational awareness, while Mixed Initiative Team Performance Assessment System (MITPAS) is widely used for human-robot team performance.

4.2.1 Operator Multitasking Performance

In a task-network computer model developed by Army Research Laboratory (ARL) [43] researchers simulated a combat environment, where a soldier has to do multiple tasks: monitoring UAV feed, controlling an autonomous reconnaissance vehicle (ARV), viewing ARV data, and providing ARV security [44]. Before multitasking, gunners were recorded having less workload, because predominantly their job is to scan targets. But, when they were assigned the task of controlling ARV, their performance deteriorated rapidly.

Chadwick et. al. used a video game to simulate the collaboration of a human controlling multiple robots [45]. In that study they found that it is really difficult for humans to switch between tasks. To facilitate multitasking with robots, tasks needed to be revised. For complex autonomous commands, participants tend to avoid those commands if they generate unpredictable behavior or are too complex to understand. For navigation, most of the participants lost track of one of the robots while working with two in a single display. In dual display, they could keep track of both robots.

Military reconnaissance UAVs typically comprise two types of operators: one for controlling airframe, and another for payload sensor control [46]. For search and rescue operations, we need to manage these two tasks. Researchers have shown that assignment of both tasks to a single operator with conventional UAV control display can substantially reduce performance [47]. As a result, we need to redesign the control interface with multitasking in mind. Also, it is important to understand crew communication [48] and focus on inter crew communication [4]. One way of implementing multitasking is to incorporate multi-sensory input/output to the control interface, such as tactile and auditory feedback. Moreover, there are no standards for selecting and training teams of UAV operators. Therefore, it is important to incorporate user studies in the design and development of UAV controllers.

4.2.2 Single vs. Team Performance

Robots will be required to accompany humans in complex and demanding tasks for disaster mitigation operations. For human-human collaboration, each participant plays a particular role, which results in a harmonious execution of the different parts of a bigger task. To build upon this for human-robot collaboration, Nikolaidis et. al. [49] used a cross-training method. In cross training, team members switch their roles with one another which results in better understanding of everyone's job. They also introduced a mental model that allows the robots to coordinate their actions with humans in a collaborative task. A comparison between the cross-training method and reinforcement learning techniques showed that the former yielded better performance in robots and increased trust in humans towards robots. In a similar study, Shah et. al. [50] used a mobile robot to collaborate with humans in an assembling-blocks task. They designed a system named *Chaski* which scheduled the robot actions according to its human collaborator and thus minimized the human idle time by 85%.

In Urban Search and Rescue (USAR) operations, robots are often desired as there are potential dangers for human rescuers. Robots were first used in the 9/11 disaster of World Trade Center (WTC) and ever since the need for rescue robot increased. In a building collapse it is difficult for rescuers to search every corner. In nuclear plant disaster it is not safe for the human rescuers to operate physically. In a 16-hour USAR drill with teleoperated robots by Burke et. al. [51], operators spent 32% of their time for search, while 54% of the time to comprehend state of the robots and surroundings. As a result, robots were stationary around 50% of the time. These findings suggest that operators have a great deal of difficulty in tele-kinesthesis and tele-proprioception with robots, resulting in lower situational awareness. They suggested a new mental model to bridge the cognition gap by filtering and pre-processing data from the robots. Murphy [1] proposed a domain theory comprised of two parts for this problem: 1. A work-flow model to identify the tasks, actions, and roles for each member of the human-robot team. 2. An information flow model to integrate data from various team members.

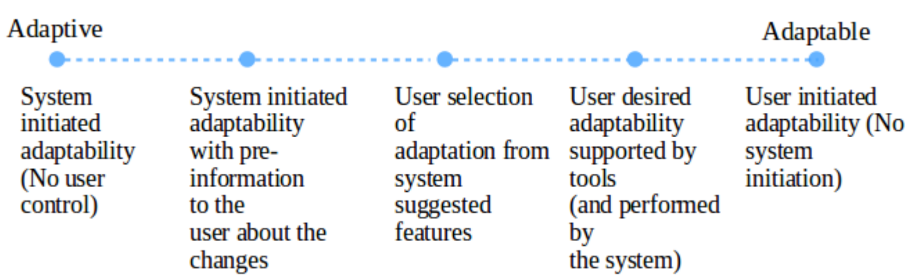
It is difficult to find what makes an effective human-robot team. For a team comprised of only humans, Bell and Cooke [52] conducted a 2-by-2 study. They found a correlation between verbal working memory and grade point average (GPA) vs. team and role performance. Participants in this study were university students with supposedly no UAV or aircraft piloting experience. Verbal working memory of individual participants was measured using the Air Force CAM 4 computerized test [53][54].

The results of the experiment showed that verbal working memory was highly correlated with role performance while GPA was indicative of better team performance. Though it is impossible to measure the verbal working memory or GPA of a robot, the result of their experiment can be extended to comprise a human-robot team.

4.2.3 Adaptability

Adaptability can be divided into two types: *adaptive* and *adaptable* systems [55]. According to Oppermann [55], in an adaptable system the flexibility of controlling information and automation such as tuning the system parameters resides in the hand of the user. Whereas in an adaptive system, the system tunes the parameters and changes the environment itself based on the user data. The following figure by Oppermann et al. [56] compares between adaptive vs. adaptable system:

Figure 4.1: Spectrum of adaptation in computer systems



The words "adaptive system," "adaptive user interface," and "adaptive automation" are widely used currently, and their meaning corresponds to the definition by Oppermann [57]—systems have the flexibility of controlling system parameters based on user performance. Adaptive systems are widely used and Miller et al. [57] pointed

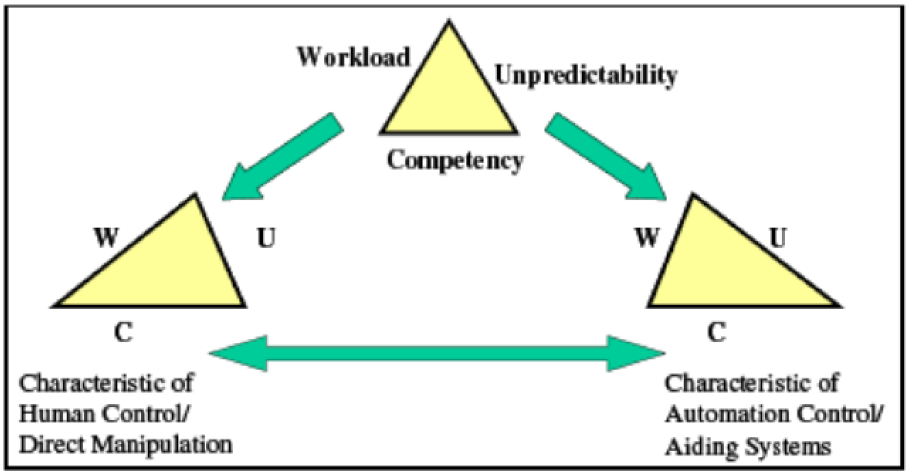


FIGURE 4.2: Performance trade-off in different level of adaptability

out some distinct advantages and disadvantages of this system. Adaptive system tend to have greater speed of performance, reduced operator workload, less training time, more flexibility in behaviors, and more consistency by effectively reducing control task from human agents. On the other hand, over reliance on adaptive systems produces unwanted results. For example, a fully adaptive system can reduce human engagement from control and decision making, thus decreasing situational awareness. It can also increase over-reliance on a system, which might result in complacency, skill degradation, etc. Finally, a fully adaptive system can result in an arise unbalanced mental workload and decreased situational awareness or user acceptance [57]. Also, to program adaptability in a system is a complex task, and requires more time and cost. Miller et al. [57] showed the trade-off among workload, unpredictability, and competency as three sides of a triangle in Fig. 4.

According to Oppermann et al. [56] a learning system takes input from the user and acts according to the inputs. Controlling a UAV can be regarded as a learning system. In traditional Intelligent Learning Systems (ITSs), the system temporarily applies some restrictions and recommends pedagogical strategies to the user [58]. The system evaluates a user's behavior and learning outcome through tests. In the case of UAV missions, researchers used sudden question/answer session while the operators were in the middle of a mission. This was done to measure the situational awareness and operator workload. When an intelligent learning system is used, the system will collect real time user data through direct questions or machine learning approaches [56]. The system then uses adaptive learning to tune its parameter to set the difficulty level. However, fully adaptive systems can degrade performance. Some system parameters should be controlled by the user. For example, audio level, navigational panel, camera focusing etc. should be adaptable.

4.2.4 Situational Awareness

When discussing the situational awareness of the human operator, the most commonly cited definition is the one given by Endsley [2]: "The perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future."

Based on the SA definition and Figure 4.3 we can say that measuring the situational awareness is equivalent to measuring the operators ability to perceive relevant information in the environment to integrate the data in conjunction with task goals, and to predict future events and system states based on this understanding.

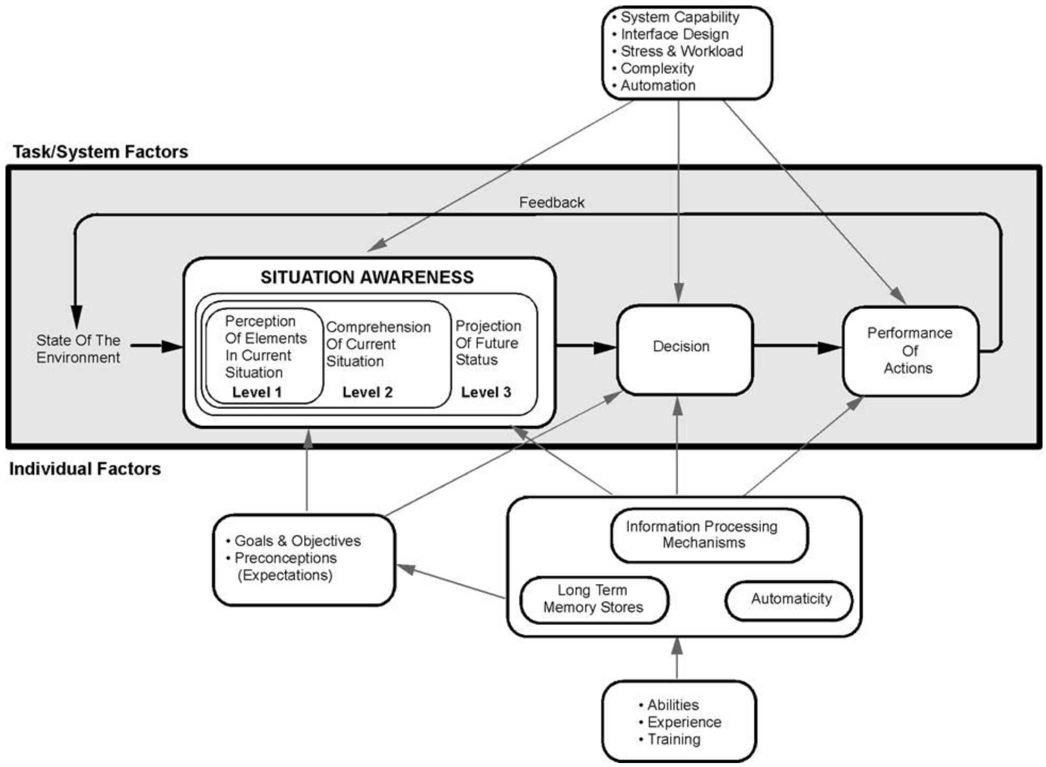


FIGURE 4.3: Situational Awareness, Endsley [2]

The most common objective measure for SA is based on Endsley's definition and is called *Situation Awareness Global Assessment Technique* (SAGAT). In SAGAT, the simulation is paused and the display is blanked while questions regarding the situation are asked. Once a participant answers all the questions, the simulation is resumed only to be stopped again at some later point for additional SAGAT questions. The level of SA is measured as a number of correctly answered questions and the time taken to answer the questions [2].

Visser *et al.* proposed the concepts of *mission model memory* and *deviation detection* as objective measurements for interface usability. The goal of this approach is to asses if the participants can remember which tasks were executed, which agent was responsible for each tasks, and how the tasks were related, which would in turn serve as an indicator of their situational awareness [59].

On the other hand subjective measures are SA measurement techniques that aim to evaluate people's self-assessment of the SA [60]. In the past the SA has been measured using techniques as direct interviews with the human operators [10]. One technique for measuring the operator's perceived SA is the *Situation Awareness Rating Technique (SART)*. The SART questionnaire requires participants to rate demand on attentional resources, supply of attentional resources and understanding of the situation on a 1-7 scale. Responses to the SART result in a subscale for each of the aforementioned dimensions as well as a combined score based on the difference between attentional demand and the sum of supply and understanding ratings [61].

4.3 Summary

We reviewed the factors associated with human-UAV interaction and interface design. We divided this spectrum into two broad categories, yet we know that there are some metrics which cannot be classified in that manner. There has been a lot of research on the human-computer interaction. As for human-robot integration, especially where a flying robot is in supervised control by an operator, there have not been as many studies of operator/interface performance. Each of these factors we discussed has a significant effect on supervised UAV control.

Chapter 5

UAV Simulator

A review of mobile robot simulation environments reveals that simulation is becoming an increasingly important aspect of mobile robots [14], helping researchers perform more experimentation in this area. A realistic graphical rendering system and ideal physics simulations are the main features of a best simulator. Computer video games engines are often used to power a robot simulation environment, capable of simulating multiple robots, people, and objects in the environment. In this chapter, we introduce a simulator developed by the Evolutionary Computing Systems Lab (ECSL) and Robotics Research Lab (RRL) at University of Nevada, Reno to experiment the design of user interface as well as the workload and situational awareness vs. level of automony for swarm UAV control.

A more recent paper proposed a UAV-based solution to help on the search and rescue activities in disaster scenarios [62]. These UAVs are specialized to perform operational tasks (e.g., providing a temporary communication structure, creating up-to-date maps of the affected region and searching for hot spots where the rescue teams may have more chances of finding victims) and attain search-and-rescue objectives. These robots utilize sensors fixed on the UAVs, such as infrared cameras, radars, or portable devices for detecting radio signals [62]. All of these activities require specific competences, and as such, more than one UAV or sensor type may be required to accomplish all of them. This UAV-based fleet, to be efficient and useful in the terrain needs to be semi-autonomous and more capable of self-organization.

Simulation is an important step before deploying a system. To use UAVs for search and rescue operation, operators must be aware of their duty and up to date with the real world state. Disasters can be very dissimilar from each other. So we have to make sure that the training of the UAV operator along with the emergency personnel gets as close as to the reality. As a result, the Robotics Research Lab and the Evolutionary Computing Systems Lab teamed up to create a real-world simulation of a disaster response mission using multiple UAVs [63].

In this simulation we developed a model of the city of Reno after an earthquake, where a number of building had fallen over. We then used UAVs to detect cars and humans in the disaster. The simulator allows an operator to fly up to 4 UAVs at a time. We conducted a simple experiment regarding operator workload and situational awareness based on different level of autonomy using this simulation. The simulator has been augmented to mimic the waypoint navigation control of a of a Hummingbird quadcopter. In the end, using a *Genetic Algorithm* [3] the simulator can optimize the path between different points in the map.

5.1 Simulator Design

The simulator was developed to simulate the operator interface for real world multi-UAV control. In earlier chapters, we showed that people prefer semi-autonomous UAVs more than fully autonomous UAVs. For that reason, in this simulation we kept the flying part automated. But navigation, path planning, collision detection were not automated. Although, total distance was optimized using automation.

Our simulation *RenoRescueSim*, was developed in Unity3d game engine, version 5.3. The terrain in the simulation was rendered directly from Google Maps. We built a earthquake ravaged city on top of that map. The model of the buildings resemble the actual buildings in the city Reno, but they simulated destruction by earthquake. The UAVs models resemble the Hummingbird quadcopter using two cameras. One front-looking camera to look in the direction of flying and another camera beneath the drone to see and tag people (see Figure 5.1). An operator can observe the world through the bird's eye view (Figure 5.4). The two cameras in the UAVs also render in 4 mini screens.

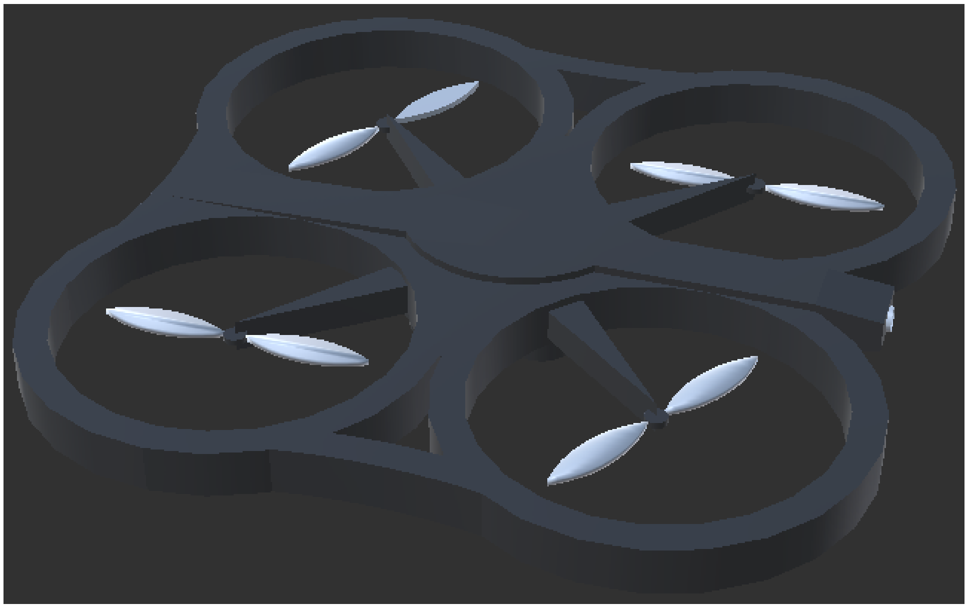


FIGURE 5.1: UAV model used in the simulator. This UAV has two cameras. One for looking directly in front and other for looking below. It also simulates all the sensors and actuators the hummingbird robot has (e.g., GPS, wireless communication, and inertial guidance system).

5.1.1 Simulator Architecture

The architecture of the simulator was designed such a way that it allows a supervisor to manipulate the simulation using *Configure Manager* (see Figure 5.2). Another principal component of the system is *Scene Manager*. The scene manager handles all the movements of the UAVs, cars, people, and helicopters; thus handles the rendering of the city terrain. Using an *XML config* file, a supervisor can control the scene manager.

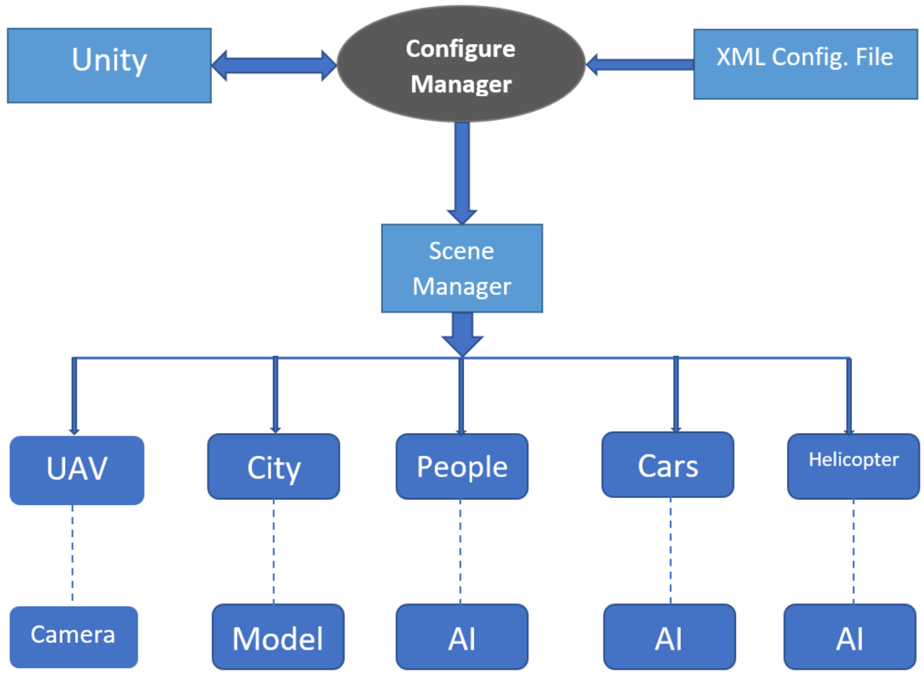


FIGURE 5.2: System Architecture. The simulator developed in Unity game engine. The configure manager can be changed using XML configuration file using Unity. The Configuration manager changes Scene manager. The scene consists of a UAV, which has two cameras. The cameras are used to render the view from the UAV. The city is modeled, except the map , which is rendered from Google Maps. There are numbers of random people, cars and helicopters roaming in the city. Their movements are controlled using AI

5.1.2 3D Environment Modeling

The Unity game engine is very popular for its availability of assets. Also, it is free to use for personal project, and there are tons of free models to use. The physics engine can adequately simulates gravity and collisions effectively enough to run on most consumer computers.

For our simulation of UAV flying, we used the gravity of Unity game engine, rather we used the gravity constant mentioned by Michael et. al. [15]. As mentioned earlier, we used Google Maps data to load the terrain of the city. We developed building models on top of the terrain to simulate the earthquake aftermath (see Figure 5.3).

The UAV we simulated is the Hummingbird quadcopter. The physical properties of the UAV can be found [15]. We combined the specific parts of the UAV from different models. We used separate models of: propellers, cameras, body frame, motor, actuation, gyros, GPS, and battery (see Figure 5.1).

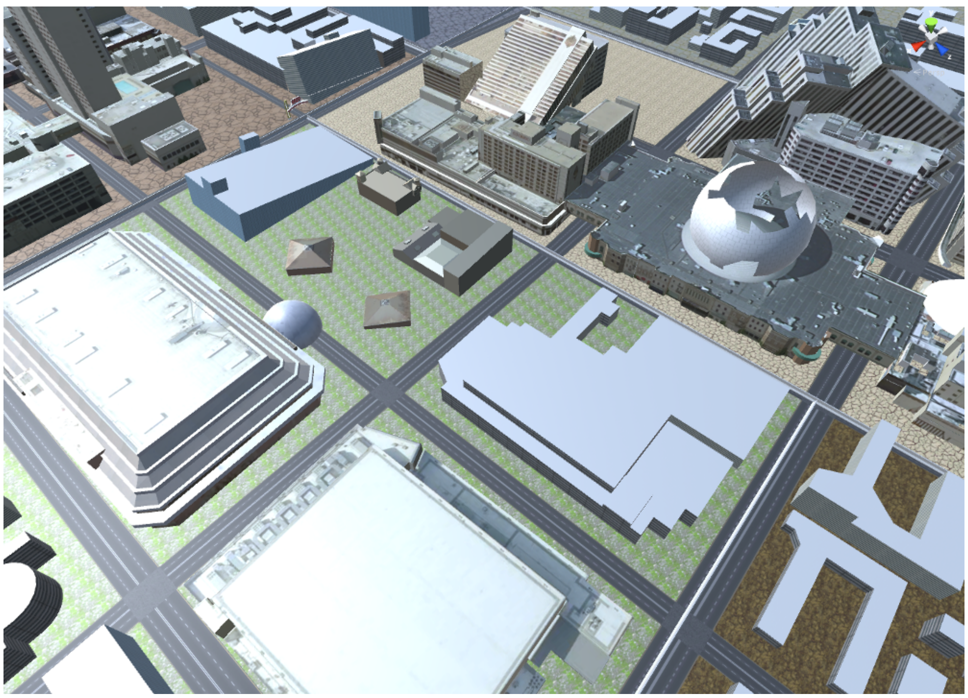


FIGURE 5.3: Simulation rendering of a damaged Reno downtown area in the Reno Rescue Simulator. This matches the actual layout of the city of Reno, but some buildings have been damaged and collapsed.

5.1.3 User-Interface Design

As we described earlier, the objective of this simulator is to measure the awareness and workload of the operator controlling UAVs in disaster response. For this reason, the user interface plays a crucial role. We have tested different UI designs and chosen the design depicted in Figure 5.4.

The display window is divided into several different small windows. The bottom left corner (see Figure 5.5) is the mini map. This mini map window renders the simulated area directly from Google Maps. The bottom-center window is divided into three segments: *selection bar*, *info bar*, and *order bar*. The selection bar shows the number of UAVs available for the mission (in this simulator, we used 4). When a particular UAV is selected, the info bar shows the information e.g, battery percentage, altitude of that particular UAV. The order bar shows the instructions that each or a group of UAVs can take. In our simulator we have: Landing, Take-off, Recharge, and Find-path. The bottom-right window renders the camera feed from a UAV when it is selected.

A single click will select a UAV. Selected UAV can be controlled via keyboard button W,S,A,D. Where W, and S moves the UAV forward and backward respectively. And, A and D rotates the UAV counter-clockwise and clockwise respectively. UAVs cannot move side wise when manually controlled. To select multiple UAVs at once and command collectively, uses draws a window around them by left-clicking. After selection, a single right-click will command the UAV/UAVs to go to that point (Figure 5.6, Figure 5.7).

The top bar of the main display is called *Resource Bar*. The resource bar shows the Day/Night toggle mode, battery life, altitude, and speed of a single UAV when

Figure 5.4: *RenoRescueSim* user interface. **Center**: is the main view, where a user can get a top-down view of the world (rendered from Google Maps tiles); **Bottom-Left**: minimap view of the entire city, with area viewable in the center panel shown (blue-box); **Bottom-Right**: View from currently-selected UAV's camera; **Top-Right**: Views from all UAV's cameras

selected. On the right of the screen, there are 4 mini-camera feed window shows real-time camera-feed from 4 UAVs. When double clicked on any of these mini window, that UAV is selected, and the bigger camera window shows the feed from that UAV.

The center of the display shows a birds-eye view of the city. This image comes directly from Google Maps, and we modeled the earthquake affected installations on top of that terrain. The center of the screen can be zoomed in/out using the mouse scroll button. To zoom quickly a user will press shift button and simultaneously scroll the mouse button. When the user touches the left/right edge of the screen with the cursor, the central window moves to the direction of the cursor. The clicks once in inside the bigger-camera window to tag person or car. When a person or car is tagged, they receives a red cubicle over them, which remains with them throughout

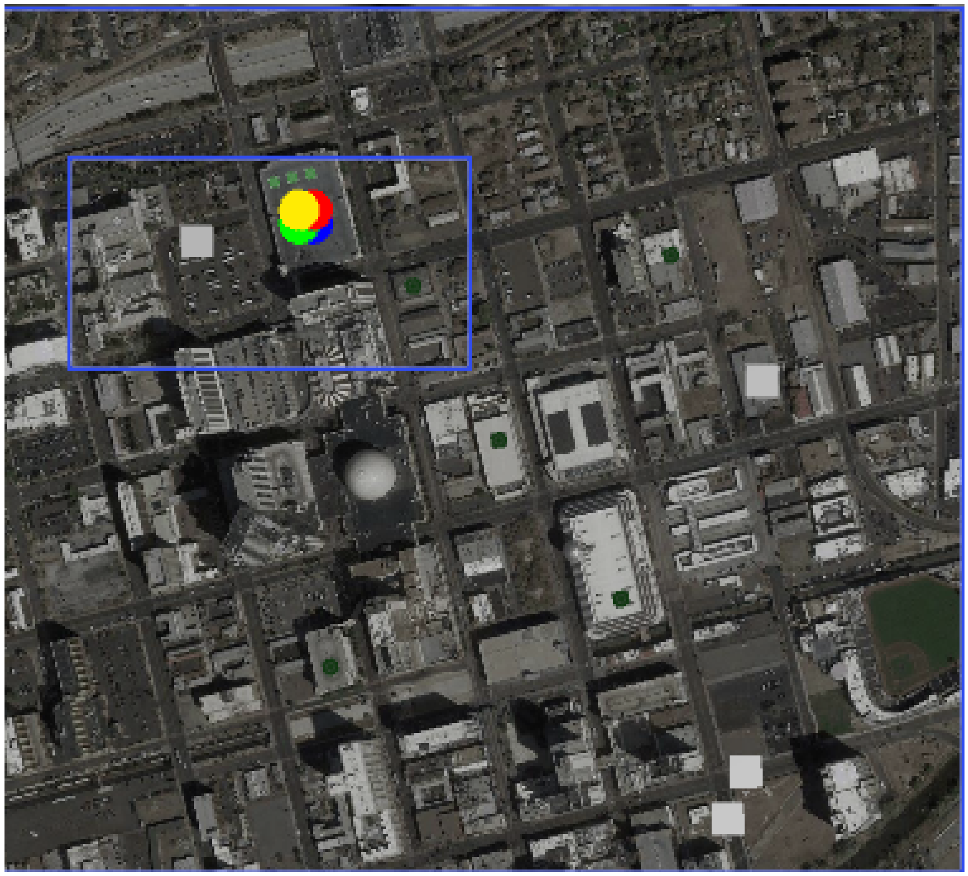


Figure 5.5: Minimap panel used to show entire map with locations of all UAVs (red, yellow, green, and blue dots). Current view of center panel is shown (blue rectangle). Users can change the center panel view or set waypoints from this panel.

the game. When a cursor is hovered onto a car or person, their size increases by 25% to facilitate this clicking.

5.2 Simulated Dynamics for Full Autonomy

To address the realism of UAV movement, we turn to established models of UAV dynamics. Michael et. al. [15] have provided an accurate aerodynamic model of micro UAV (MAV) flying. MAVs are between 0.1-0.5 meters in length, and 0.1 to

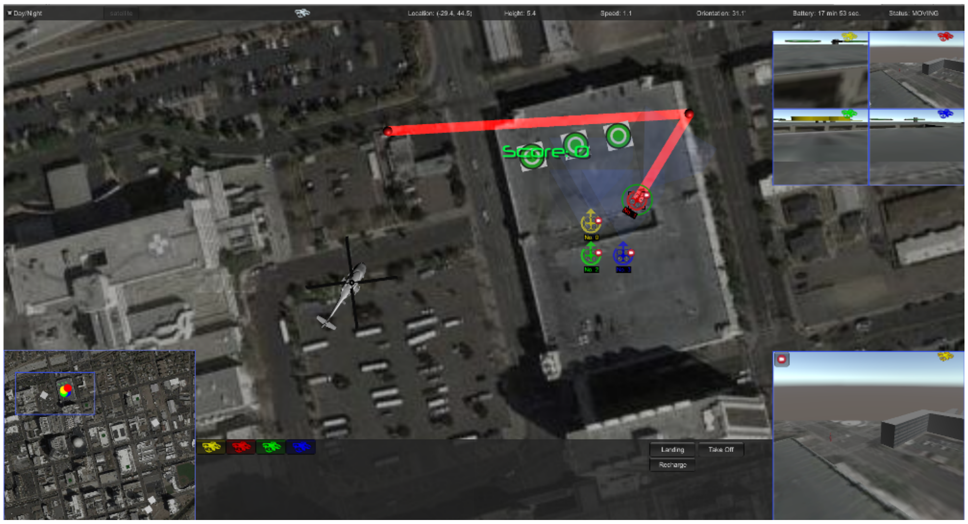


FIGURE 5.6: Single-UAV navigation with two sequential waypoints set (red lines) (Level 2, 3 autonomy cases). Waypoints can be set by selecting a UAV and right-clicking either in the center panel or in the minimap view.

FIGURE 5.7: Multiple-UAV navigation with several waypoints set for multiple UAVs to move in formation (Red, Yellow, Cyan, Blue lines) (Level 3 autonomy).Waypoints can be set by selecting multiple UAVs and right-clicking either in the center panel or in the minimap view.

0.5 kilograms in mass [64]. MAVs are commonly utilized for civilian applications; therefore, they are the size class platform that we will simulate for this work.

Michael's model is specifically designed for the Hummingbird quadrotor sold by Ascending Technologies. It has a 55 cm tip-to-tip wingspan, 8 cm height, and 500 grams of weight including the battery. Also, it has a battery life of 20 minutes, and can carry 200 grams of payload [15]. The small size and dexterity of Hummingbird UAV made it suitable to navigate through a constrained space. In this section, we implemented the aerodynamics of the Hummingbird UAV using the formulae by [15].

To make the UAV movement more realistic and suitable for training purposes, some critical aspects of UAV movement need to be considered. First, a UAV will not move from point A to point B at a uniform rate as the dynamics of the system need to be considered. Furthermore, as the UAV changes velocity in any direction, pitch and roll changes occur. As it is likely that a fixed camera on a UAV will be what an operator will use for a search-and-rescue task, simulating such attitude changes would be crucial for an operator's later proficiency with a real-world system.

We make our UAV's simulated flight path mimic the state of a UAV for real-world flight. Three axes x, y and z, that locate its position, and three angles ϕ, θ, and ψ that measure the angular distance from respective axes are derived from this model. These variables control the movement and orientation of a UAV. Each of these variables are a function of the angular speed of the rotors. The angular speed of each rotor creates thrust and lift, which are opposed by the forces due to drag and gravity. Though the effect of wind is significant for the movement of such a small aircraft, we discarded the effect of wind in this simulation for the sake of simplicity. We assume that, the

UAV will be flying in a closed environment where the effect of wind is nominal.

When an operator provides a series of points for the UAV to follow, the simulated controller will plan a trajectory to reach each goal, obeying the dynamics of the system. The UAV updates its flight path by using a Proportional Derivative (PD) controller. The movement and orientation of the UAV showed in the simulator represents real world UAV flying. This results in the simulator tilting while turning and pitching when accelerating/decelerating, which resembles a real world UAV flight. It also sets a more dynamically appropriate trajectory than a carrot-style planner. We wanted to simulate real world UAV flight for the purposes of training so that rescue operators would have a solid understanding of how such a system would move during emergency operations.

We implemented the flight dynamics of the UAV in two steps. In the first step, we defined the physical properties related to the UAV flying. Those are: mass of the UAV, gravitational acceleration, thrust co-efficient for motor, distance from the center of the UAV to the rotors, PD control parameters (for controlling position and orientation), and moment coefficient for motors.

Next, for each frame in the Unity game engine, we calculated the desired angular speed, rotational speed, attitude control parameters, force, moment, and inputs for each rotor, net force acted upon the UAV, and orientations (yaw, pitch, and roll angles) with respect to three axes. We compensated the error of the UAV from the desired flight path by using the PD parameter and added that error in every frame.

Each frame in Unity represents the minuscule time interval dt. We used 60 frames per second for this simulation [63].

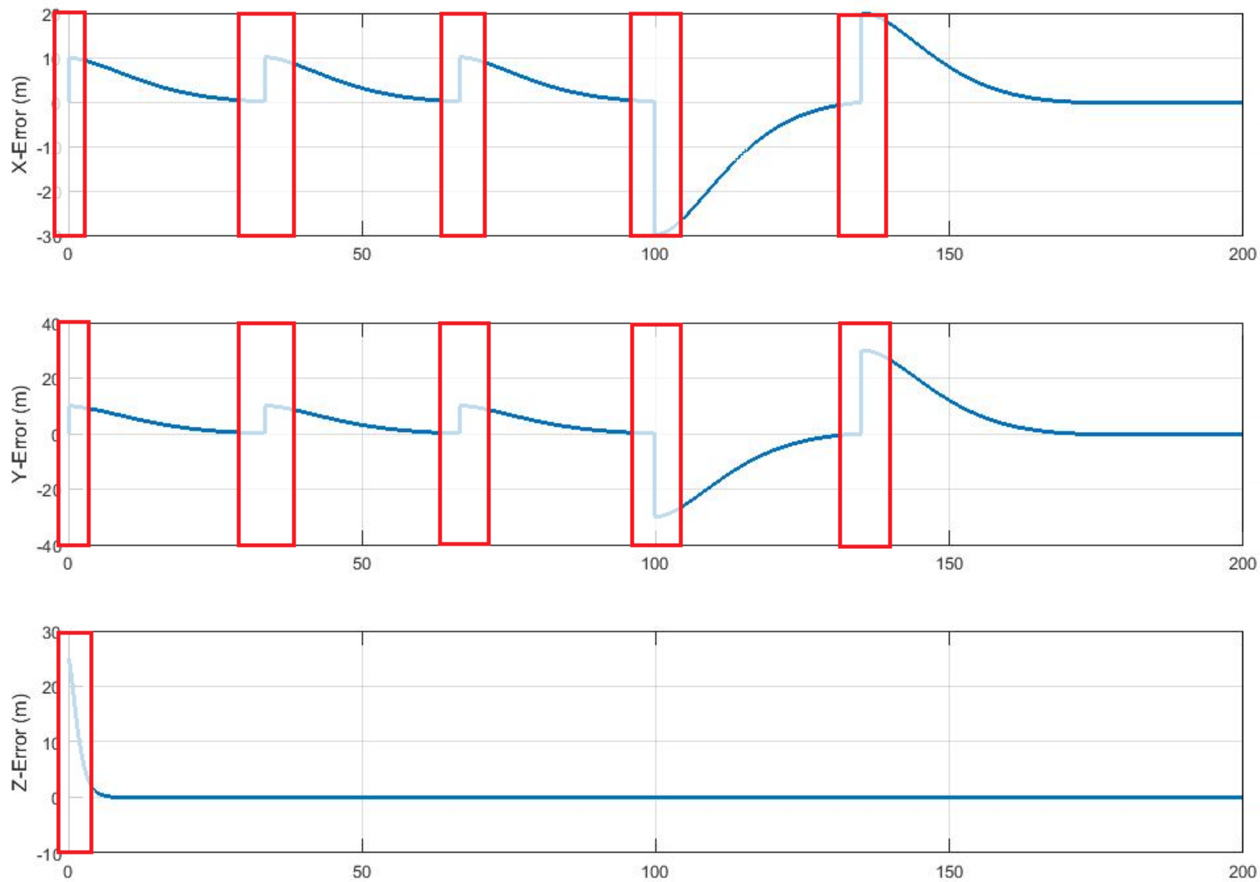


FIGURE 5.8: Graph showing the difference in planned trajectory and current trajectory (waypoint transitions are indicated by red rectangles)

The simulation showed us how the UAV followed the flight path created by the autonomous flight dynamics algorithm. We show an example movement in Figure 5.9. Our simulator showed that the UAV (brown) followed a flight path and is able to repeatedly reached the destination. The flight path created by the UAV was reasonable and quick to implement. The flight path was direct and slowed its velocity when

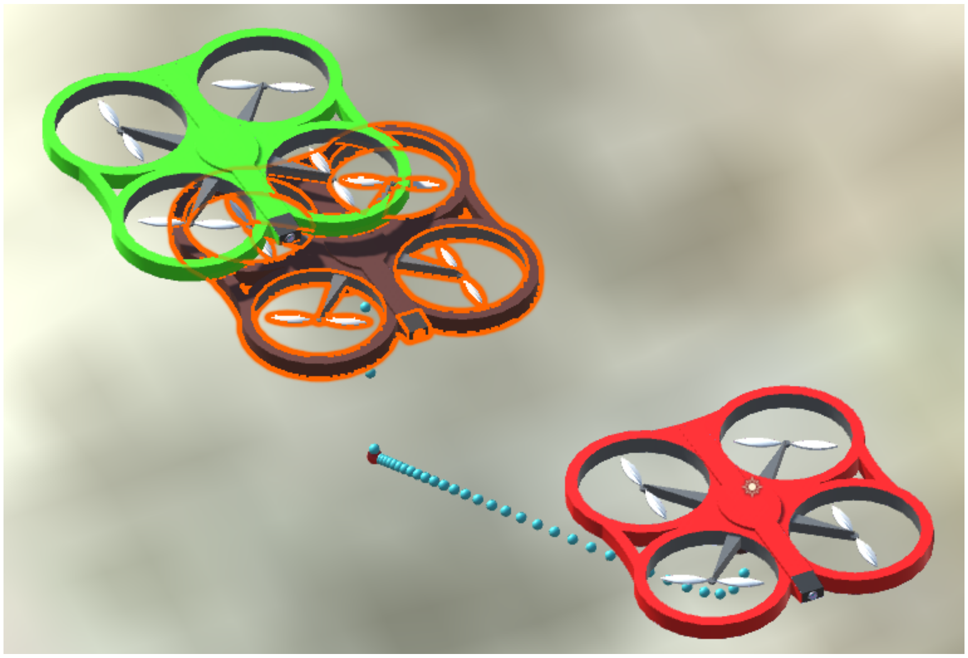


FIGURE 5.9: Modified trajectory planner in action. The red UAV represents the start position; the brown UAV is the current position (trajectory shown with blue dots) and green UAV indicating the goal.

it was close to the destination (see Figure 5.8). Upon reaching the destination, the UAV hovered to maintain its position and orientation.

5.3 Distance Optimization

Optimizing the total distance travelled is very crucial for UAVs. If we have to travel through N different points, then the order of visiting those points is very important [65]. Because, crossing the path while travelling or visiting a point more than once would cause us more time and energy. In a disaster response scenario, it is not advisable.

Genetic Algorithm [3] is a popular way of solving this problem. It can optimize the total distance in a very short time using less computation. For example, if we have 10 points, we would have $10! = 3{,}628{,}800$ possible combinations of routes. If we were to calculate the total distance of these routes and compare them with each other, the computation and comparison would require an intractable amount of time and computation power.

This Genetic algorithm we use string *chromosomes* to represent a particular sequence of waypoints. We then crossover and mutate the chromosomes with each other to find new generation of chromosomes. Gene is called a particular portion of the chromosome in GA. While crossover or mutation, the GA keeps replaces the weak genes in the new generations with stronger genes. In this example, gene represents a small set of locations. If a particular gene has greater total distance than another gene, we call that gene *weak*. And that weak gene is replaced in the crossover of mutation phase.

One of the most common problem for optimization is the local minimum. We mutated up to 10% the chromosomes in a generation. For crossover, if there were no new chromosomes after crossover, we forced a mutation. We initialize the GA with 150 population and ran for 500 epoch.

Algorithm 1 Mutation Type-I: Swap Location

if $(index1 < 0) or (index1 \geq locations.length)$ **then**
    $ThrowOutOfRangeError$
end if
if $(index2 < 0) or (index2 \geq locations.length)$ **then**
    $ThrowOutOfRangeError$
end if
$location1 \leftarrow locations[index1]$
$location2 \leftarrow locations[index2]$
$locations[index1] \leftarrow location2$
$locations[index2] \leftarrow location1$

Algorithm 2 Mutation Type-II: Move Location

if $(fromIndex < 0) or (fromIndex \geq locations.length)$ **then**
    $ThrowOutOfRangeError$
end if
if $(toIndex < 0) or (toIndex \geq locations.length)$ **then**
    $ThrowOutOfRangeError$
end if
$temp \leftarrow locations[fromIndex]$
if $fromIndex < toIndex$ **then**
    for $i \in toIndex$ **do**
        $locations[i-1] \leftarrow locations[i]$
    end for
else
    for $i \in toIndex$ **do**
        $locations[i] \leftarrow locations[i-1]$
    end for
end if

In this GA we presented three different types of mutations. First one is *Swap Location* [66]. This swaps random location with each other, not just change bits/location like

Algorithm 3 Mutation Type-III: Reverse Range

if $(startIndex < 0) or (startIndex \geq locations.length)$ **then**
  $ThrowOutOfRangeError$
end if
if $(endIndex < 0) or (endIndex \geq locations.length)$ **then**
  $ThrowOutOfRangeError$
end if
$temp \leftarrow locations[fromIndex]$
if $endIndex < startIndex$ **then**
  $temp \leftarrow endIndex$
  $endIndex \leftarrow startIndex$
  $startIndex \leftarrow temp$
end if
while $startIndex < endIndex$ **do**
  $temp \leftarrow locations[endIndex]$
  $locations[endIndex] \leftarrow locations[startIndex]$
  $locations[startIndex] \leftarrow temp$
  $startIndex + +$
  $endIndex - -$
end while

usual GA (see Algorithm 1). Second type of mutation is called *Move Location* [66], which moves one location to another position and rearranges the whole combination of the waypoints (see Algorithm 2). The third one is *Reverse Range* [66], which reverses part of the chromosomes (see Algorithm 3). For Crossover [66], we took two random chromosomes in a generation until there were no leftover for crossing. We picked up a random point in between zero and the chromosome length. Then we swapped the rest of the chromosomes with each other. In this case, we confront a problem which is duplicate locations in a chromosome. To solve this problem, we first identify the duplicate locations. We then replace this locations from the pool of unused locations for each chromosome (see Algorithm 4). After crossover, we did *Selection* [66]. Selection divided the population in half and makes another set of

Algorithm 4 Crossover

```
startPosition ← GetRandomValue(locations1.Length)
crossOverCount ← GetRandomValue(locations1.Length − startPosition)
for i ∈ crossOverCount do
    locations2[i] ⇔ locations1[i]
end for
index ← 0
for value ∈ locations1 do
    if !availableLocations.Remove(value) then
        if toReplaceIndexes == Null then
            toReplaceIndexes = newList < int >
        end if
        toReplaceIndexes.Add(index)
    end if
    index + +
end for
if toReplaceIndexes! = Null then
    enumeratorIndex ← toReplaceIndexes.GetEnumerator()
    enumeratorLocation ← availableLocations.GetEnumerator()
    while true do
        if !enumeratorIndex.MoveNext() then
            break
        end if
        if !enumeratorLocation.MoveNext() then
            ThrowInvalidOperationException
        end if
        locations[enumeratorIndex.Current] ← enumeratorLocation.Current
    end while
end if
```

new generation. Selection chooses the best N number of chromosomes from the *2N* number of chromosomes, and discard the rest.

The result of this Genetic Algorithm working perfectly is shown in Figure 5.10. In this figure we demonstrated how the GA worked found the shortest path for 18 waypoints. When we randomly click in the simulator, the GA takes those coordinates as inputs and finds the shortest path. The Figure 5.10 shows that there were no overlapping of the path and no point were visited twice. This observation proves that, we found the shortest path.

FIGURE 5.10: Figure showing the shortest path found by the GA for 18 waypoints. Smaller Aqua points are the waypoints the UAV has to visit. Bigger red points have been already visited by the UAV.

5.4 Summary

In this chapter we described about the simulator we developed in our lab. We also briefly described the necessity of the simulation for search and rescue operation. We also presented our architecture of the simulator. We believe this simulator will be helpful to integrate disaster response team with robots. This simulator was also helpful to understand the workload and awareness while in a disaster mission. We described how each part of the simulator was designed and how different parts worked together to make a real world simulation. The flight dynamics of Hummingbird quadcopter was integrated in the simulator to provide a better understanding of the physical property of the UAV. Genetic Algorithm was used to optimize the total distance among the way points. Our future work regarding this simulator will be to relay live video feed from real-world UAVs.

Chapter 6

Measuring Awareness & Workload

We set up an experiment using the UAV simulator, to see the correlation between level of autonomy versus operator workload and situational awareness [63]. Our initial validation of the system addresses two research questions. First, is a RTS (Real-time strategy) style interface effective for controlling multiple UAVs? Second, if a person has prior experience playing a RTS game, does it affects his/her performance in our simulation of search and rescue operation?

We divided the autonomy level of the UAVs into three categories. They are:

- **Level 1**: an operator can only control one UAV through direct flight control (increase/decrease altitude, move forward, backward, turn in place, slew left-/right).

- **Level 2**: an operator is able to control one or more UAVs either through direct controls or by setting a single destination waypoint.
- **Level 3**: an operator is able to set one or more waypoints for a UAV to follow in sequence.

We performed a 3x3 between- and within-subjects study with two factors: autonomy type and trial number. Autonomy type has three levels (described above). Trial number has three levels: one, two, and three. We examined the simulator behavior using several dependent variables: situational awareness, mental demand using the simulator, physical demand, and frustration.

To measure situational awareness, we asked users after each 5-minute trial to answer questions related to the health and location of their UAV fleet. We asked users to estimate the battery level left (the level starts at 100, and decreases based on the amount of movement and time in the air, which can be regenerated by navigating back to the "home base" for the UAVs). The actual battery level is compared to the estimated level to get an accuracy measure.

We used the NASA Task-Load Inventory (TLX) [67] to estimate a user's mental and physical demand as well as their frustration with the interface after each trial. This is a well-established scale to measure an operator's effort when completing tasks, and has been applied for many general problems, especially user interfaces.

We hypothesized the following:

- **H1**: Proficiency in the search-and-rescue task will increase the more an operator uses the simulator (practice effects). This can be measured by comparing operator situational awareness changes over time (higher is better) and by comparing the mental and physical demand of using the simulator (lower is better).
- **H2**: The robot autonomy type (described above) will affect user demand and frustration with the interface (lower is better). The users will perform better with greater autonomy.

6.1 Participant Recruitment

Fifteen undergraduate and graduate students (10 Males, 5 females) with no prior experience with rescue operations or simulations of UAVs were recruited from the department of Computer Science and Engineering, University of Nevada, Reno. We recruited them by sending an email to invite them to participate in the experiment. Interested participants signed up for a 45 minute time slot. The participants' age ranged from 17 to 25 years. All are regular users of computers. Most of them were familiar with playing on-line computer video games.

6.2 Experiment Procedure

After welcoming the participants the experimenter gave some basic information about the purpose of this study asked them to sign a consent form. Prior to the experiment, participants completed a pre-test questionnaire soliciting demographic data, computer expertise, and familiarity with video games. The experiment began with a training session to acclimate the users to the simulator and its operation and the search and rescue goals. The experimenter demonstrated how to move the UAV to various locations and also how to play the game by locating the people and the cars and the scoring system. The training session was followed by actual experiment. The NASA Task Load Index (TLX) as well as a situational awareness questionnaire are presented to the operator 3 times, once after each 5-minute trial assessing the operator's awareness of the scene and the UAVs they are controlling. Participants were asked to accomplish the game tasks quickly and efficiently.

6.3 Experiment Setup

The experiment was performed in the ECSL (Evolutionary Computing Systems Lab), University of Nevada Reno. It is a quiet room with no background noise so that participants were able to concentrate more on the game. The participants were asked to use a Windows workstation running the simulator. The computer used an Intel Core i5 processor and 16 GB RAM. The system would execute the simulator and

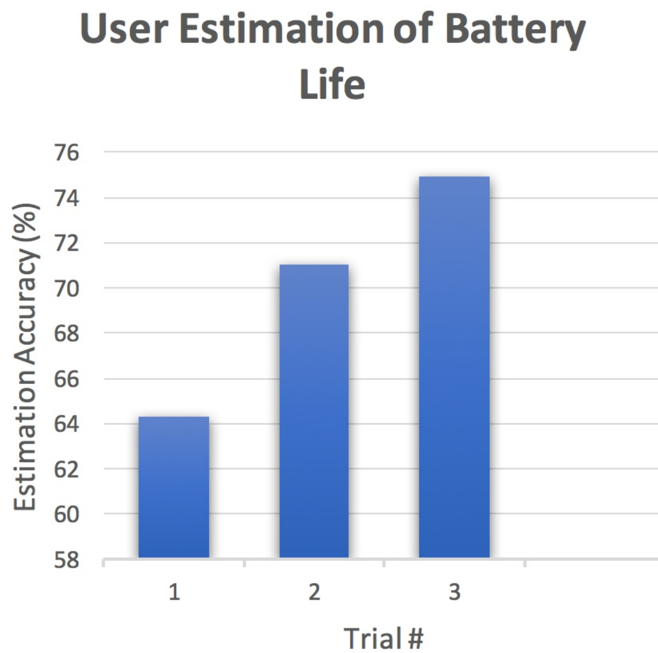


Figure 6.1: Situational awareness (higher is better) of a user by time spent interacting with the simulator expressed by the user's accuracy at estimating the remaining Battery Life of the UAV fleet.

collect data from the experiment. We collected data from each trial using the data logger built into the simulator for storage in XML files. These data included responses to the questionnaires, and in-simulation usage data (actions-per-minute, overall health of the UAV swarm)

6.4 Results

To examine hypothesis **H1**, we compared the values of situational awareness, mental demand, physical demand, and frustration for each time trial. For **H1** to be supported, situational awareness will increase and the others will decrease as more time

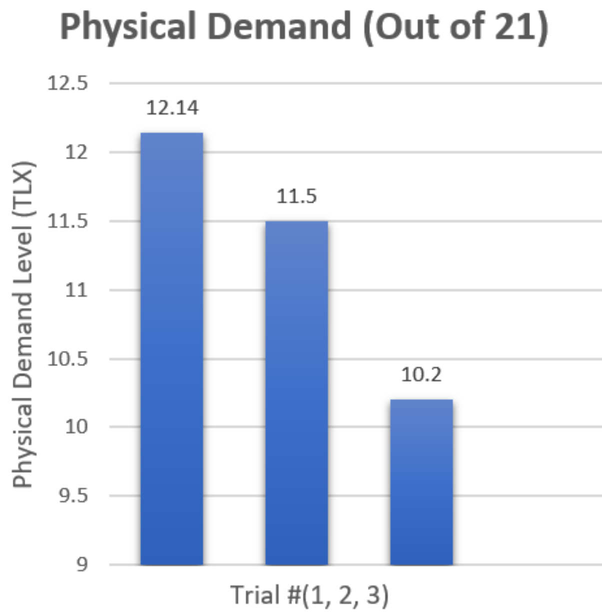


FIGURE 6.2: Physical demand (lower is better) reported by operators of the UAV after each trial (NASA-TLX survey). The decreased physical demand with each successive simulation trial indicate that training with the simulator makes it easier to use ($p < 0.001$).

is spent with the simulator.

Figure 6.1 shows the operators' accuracy estimating the battery life of UAVs during each experiment trial. The data show that as the operator gains more experience with the simulator interface, the users' accuracy estimating UAV battery life improved. This accuracy increase suggests that operator situational awareness increased with simulator practice. While these results were not significant, it is likely that a larger sample size will improve the significance of these results.

Figures 6.2 and 6.3 show the users' mental and physical demand level (measured by the TLX survey) by trial. Later rounds show less mental and physical demand was required (differences were not significant). This demonstrates that the more experience a user has with the simulator interface, their cognitive load decreases, demonstrating a training effect.

To examine hypothesis **H2**, we compare the same factors with autonomy level as the independent variable. We conducted a MANOVA with Physical Demand, Mental Demand, and Frustration as the dependent variables and autonomy level as the independent variable. The multivariate result was significant for autonomy level, Pillai's Trace = 0.43, $F = 4.46$, df = 36, $p < 0.01$. Follow-up univariate tests showed that Frustration was significant, $p < 0.001$ and Mental Demand was marginally significant, $p = 0.058$. Tukey's HSD tests showed that Levels 2 and 3 were significantly lower than Level 1.

6.5 Summary

These data partially support hypothesis **H1**, showing that there is a trend in the direction pointed to by the hypothesis, but not enough to conclude that there is a training effect due to the simulator. It is likely that given more time, and a larger participant pool, the data would show a greater training effect.

These data support hypothesis **H2**. Lower mental demand and frustration were observed when the robots behaved with more autonomy. These results make sense, since a user was able to more easily operate the UAVs while also performing the

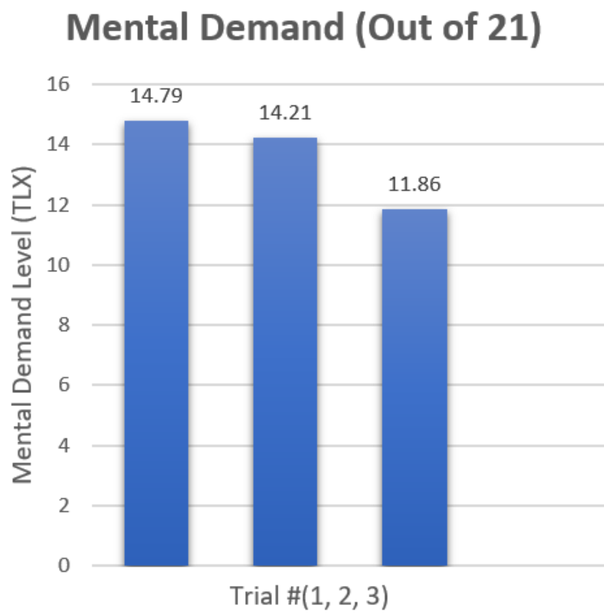


FIGURE 6.3: Mental demand (lower is better) reported by operators of the UAV after each trial (NASA-TLX survey). The decreased mental demand with each successive simulation trial indicate that training with the simulator makes it easier to use ($p = 0.058$).

search-and-rescue task. It is likely that given a greater simulator time, the users would have as high or higher differences between the autonomy groups.

While these results are promising for the use of such as a system as a training simulator (**H1**) and to evaluate elements of UAV user interfaces (**H2**). As part of our collaboration with UAV researchers, we identified several areas where the UAV did not perform as accurately in simulation to what real-world behavior would be. As this could have significance on the training value of such a simulator, we wish to increase the realism, particularly of the UAV movement in simulation.

Chapter 7

Conclusion & Future Work

7.1 Conclusion

We provided an inclusive overview of UAVs, how to use them for search and rescue operation, how the disaster response can be simulated, how the path can be optimized, and finally how the user interface of the UAV affects an operator's workload and awareness.

Chapter 2 presented some related work regarding UAVs classification, their usage in disaster mission, justifications for the use of semi-autonomous UAVs, research on the development of user interface for controlling UAVs etc. We also included research on definition for the level of autonomy, and its effect on operator's performance. We also provided some brief discussion on the similarity between controlling UAV and playing

RTS games. Finally, we described the work related to the dynamics of Hummingbird quadcopter.

Chapter 3 discusses about various researchers' contribution to classify autonomy levels. We also presented some studies which showed the effect of autonomy level on operator's workload and situational awareness. We also presented our experiment regarding the effect of different level of autonomy users' performance. We also discussed how trust is related to the level of autonomy.

Chapter 4 reviews literature regarding the factors affecting the human-robot interaction. By discussing the factors we came to a conclusion that certain factors are more important than others while designing the user interface. A lot of researchers has contributed in this field, but very few tried to classify these factors. We divided these factors into two broad categories: Subjective and Objective. Thus we tried to answer the question how and which category of factors affect the performance of the operators. We also discussed about some measurement scale for example, NASA-TLX, SWAT, MITPAS, SAGAT etc.

In Chapter 5 we present about the UAV simulator that was developed in collaboration with ECSL and RRL at University of Nevada, Reno. We described the objective of developing the simulator, the architecture of the simulator, the modelling and AI behind the movement of the objects in the simulator, and the design of the UI. We also presented the flight dynamics of Hummingbird quadcopter, which was implemented in this simulator. Finally, we described our effort to optimize the total distance covered by a UAV in a particular mission. We solved this problem by introducing Genetic Algorithm [3], which lessen the time and computing by a far.

Chapter 6 presents an experiment where we tried to find a correlation between operator workload and situational awareness versus level of autonomy. We used the NASA Task Load Inventory (TLX) to measure the awareness of a user while they were playing the simulation. We also answered the question that, whether earlier experience of playing RTS games enhance the performance of a user. We also showed that how level of autonomy affects the Mental Demand and Physical Demand while in a simulated search and rescue mission.

Overall, this thesis thoroughly reviews UAV-UI designed for search and rescue operation. We have developed a realistic simulation that utilizes design best practices, and evaluates that system in a training capacity.

7.2 Future Work

The future of UAVs in civilian application looks very promising. The UAV market is increasing rapidly, and there are thousands of problems which can be solved using this technology. We only aimed at the usage of UAVs in disaster response in this thesis. Our future work will be to integrate a real UAV in the simulation. We are trying to make the simulation more realistic. We are also working on to relay the live camera feed from a UAV in real-world disaster scenario. Our next step will be to use UAVs in a real world search and rescue operation and see how it differs from the simulation. We would also like to see the difference in performance, workload, and awareness of the operators in the real-world versus simulated mission.

Bibliography

[1] Robin R Murphy. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34 (2):138–153, 2004.

[2] Mica R Endsley. Measurement of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37 (1):65–84, 1995.

[3] David E Goldberg. Genetic algorithms in search, optimization, and machine learning. 1989.

[4] Kevin W Williams. A summary of unmanned aircraft accident/incident data: Human factors implications. Technical report, DTIC Document, 2004.

[5] Robin Murphy. *Introduction to AI robotics*, chapter From teleoperation to autonomy. MIT press, 2000.

[6] Garth Herman, Aleksander Milshteyn, Airs Lin, Manuel Garcia, Charles Liu, Darrell Guillaume, Khosrow Rad, and Helen Boussalis. Cooperative semi-autonomous robotic network for search and rescue operations. In *International Conference on Universal Access in Human-Computer Interaction*, pages 636–647. Springer, 2014.

[7] Carlos Marques, Joao Cristoao, Pedro Lima, Joao Frazao, Isabel Ribeiro, and Rodrigo Ventura. Raposa: Semi-autonomous robot for rescue operations. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3988–3993. IEEE, 2006.

[8] Jill Drury, Laurel D Riek, Alan D Christiansen, Zachary T Eyler-Walker, Andrea J Maggi, and David B Smith. Evaluating human-robot interaction in a search-and-rescue context. In *Proc. of PERMIS*, 2003.

[9] Morgan Quigley, Michael A Goodrich, and Randal W Beard. Semi-autonomous human-uav interfaces for fixed-wing mini-uavs. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2457–2462. IEEE, 2004.

[10] Frauke Driewer, Herbert Baier, and Klaus Schilling. Robot–human rescue teams: a user requirements analysis. *Advanced Robotics*, 19(8):819–838, 2005.

[11] Michael Baker, Robert Casey, Brenden Keyes, and Holly A Yanco. Improved interfaces for human-robot interaction in urban search and rescue. In *SMC (3)*, pages 2960–2965. Citeseer, 2004.

[12] BJ Fogg, Jonathan Marshall, Othman Laraki, Alex Osipovich, Chris Varma, Nicholas Fang, Jyoti Paul, Akshay Rangnekar, John Shon, Preeti Swani, et al. What makes web sites credible?: a report on a large quantitative study. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 61–68. ACM, 2001.

[13] Christopher D Wickens. Multiple resources and performance prediction. *Theoretical issues in ergonomics science*, 3(2):159–177, 2002.

[14] F Josh, S Cheryl, and S William. A video game-based mobile robot simulation environment, intelligent robots and systems. In *IEEE/RSJ International Conference, October*, pages 3749–3754, 2006.

[15] Nathan Michael, Daniel Mellinger, Quentin Lindsey, and Vijay Kumar. The grasp multiple micro-UAV testbed. *IEEE Robotics & Automation Magazine*, 17 (3):56–65, 2010.

[16] Phillip J Durst and Wendell Gray. Levels of autonomy and autonomous system performance assessment for intelligent unmanned systems. Technical report, DTIC Document, 2014.

[17] I Low. A tool for the assessment of the impact of change in automated atm systems on mental workload. 2004.

[18] V. A. Riley. *Human use of automation.* PhD thesis, University of Minnesota, Minneapolis, MN., 1994. unpublished thesis.

[19] Mustapha Mouloua and Raja Parasuraman. *Human performance in automated systems: Current research and trends*. Hillsdale., 1994.

[20] Christopher D Wickens and Stephen Rice Dixon. *Human machine interface analysis of unmanned vehicle systems*. University of Illinois at Urbana-Champaign, Human Factors Division, 2006.

[21] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 46(1):50–80, 2004.

[22] L. Bainbridge. *New technology and human error*, chapter Ironies of automation, pages 271–283. Wiley, New York, 1987.

[23] Robin R Murphy. Proposals for new ugv, umv, uav, and hri standards for rescue robots. In *Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop*, pages 9–13. ACM, 2010.

[24] Jean Scholtz, Jeff Young, Jill L Drury, and Holly A Yanco. Evaluation of human-robot interaction awareness in search and rescue. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 2327–2332. IEEE, 2004.

[25] ND Geddes and Robert J Lee. Intelligent control for automated vehicles: A decision aiding method for coordination of multiple uninhabited tactical aircraft.

In *Proceedings of AUVSI'98 25 th Annual Symposium and Exhibition*, pages 8–12, 1998.

[26] Yei-Yu Yeh and Christopher D Wickens. Dissociation of performance and subjective measures of workload. *Human Factors*, 30(1):111–120, 1988.

[27] Sarah Miller. Workload measures. *National Advanced Driving Simulator. Iowa City, United States*, 2001.

[28] Stephen M Casner and Brian F Gore. Measuring and evaluating workload: A primer. *NASA Technical Memorandum*, (2010-216395), 2010.

[29] John J Spravka, Deborah A Moisio, and Mary G Payton. Unmanned air vehicles: A new age in human factors evaluations. Technical report, DTIC Document, 2003.

[30] Alan H Roscoe. In-flight assessment of workload using pilot ratings and heart rate. Technical report, DTIC Document, 1987.

[31] Walter W Wierwille, Mansour Rahimi, and John G Casali. Evaluation of 16 measures of mental workload using a simulated flight task emphasizing mediational activity. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 27(5):489–502, 1985.

[32] Tobias Nisser and Carl Westin. Human factors challenges in unmanned aerial vehicles (uavs): A literature review. *School of Aviation of the Lund University, Ljungbyhed*, 2006.

[33] Daniel Gopher and Emanuel Donchin. Workload: An examination of the concept. *Handbook of Perception and Human Performance, Vol II, Cognitive Processes and Performance. New York: Wiley & Sons*, 1986.

[34] Russell Hardin. *Trust*. Polity, 2006.

[35] Kevin Anthony Hoff and Masooda Bashir. Trust in automation integrating empirical evidence on factors that influence trust. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 57(3):407–434, 2015.

[36] Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. The role of trust in automation reliance. *International Journal of Human-Computer Studies*, 58(6):697–718, 2003.

[37] Jessie YC Chen, Michael J Barnes, and Michelle Harper-Sciarini. Supervisory control of multiple robots: Human-performance issues and user-interface design. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(4):435–454, 2011.

[38] Christopher D Wickens and Stephen Rice Dixon. *Is There a Magic Number 7 (to the Minus 1)?: The Benefits of Imperfect Diagnostic Automation: A Synthesis of the Literature*. University of Illinois at Urbana-Champaign, Aviation Human Factors Division, 2005.

[39] Kathleen L Mosier and Linda J Skitka. Human decision makers and automated decision aids: Made for each other. *Automation and human performance: Theory and applications*, pages 201–220, 1996.

[40] Ericka Rovira, Kathleen McGarry, and Raja Parasuraman. Effects of imperfect automation on decision making in a simulated command and control task. *Human Factors*, 49(1):76–87, 2007.

[41] Raja Parasuraman, Scott Galster, Peter Squire, Hiroshi Furukawa, and Christopher Miller. A flexible delegation-type interface enhances system performance in human supervision of multiple robots: Empirical studies with roboflag. *IEEE Transactions on systems, man, and cybernetics-part A: Systems and Humans*, 35(4):481–493, 2005.

[42] E De Visser, R Parasuraman, A Freedy, E Freedy, and G Weltman. A comprehensive methodology for assessing human-robot team performance for use in training and simulation. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 50, pages 2639–2643. SAGE Publications Sage CA: Los Angeles, CA, 2006.

[43] D Mitchell and T Henthorn. Soldier workload analysis of the mounted combat system (mcs) platoon's use of unmanned assets (arl-tr-3476). *Aberdeen Proving Ground, MD: US Army Research Laboratory*, 2005.

[44] Diane K Mitchell and Jessie YC Chen. Impacting system design with human performance modeling and experiment: Another success story. In *Proceedings of*

the Human Factors and Ergonomics Society Annual Meeting, volume 50, pages 2477–2481. Sage Publications, 2006.

[45] Roger A Chadwick, Douglas J Gillan, Dominic Simon, and Skye Pazuchanics. Cognitive analysis methods for control of multiple robots: Robotics on $5 a day. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 48, pages 688–692. SAGE Publications Sage CA: Los Angeles, CA, 2004.

[46] Jason S McCarley and Christopher D Wickens. Human factors concerns in uav flight. *University of Illinois at Urbana-Champaign Institute of Aviation, Aviation Human Factors Division*, 2004.

[47] L Van Breda. Operator performance in multi maritime unmanned air vehicle control. Technical report, DTIC Document, 1995.

[48] Jamie C Gorman, Peter W Foltz, Preston A Kiekel, Melanie J Martin, and Nancy J Cooke. Evaluation of latent semantic analysis-based measures of team communications content. In *Proceedings of the Human Factors and Ergonomics Society annual meeting*, volume 47, pages 424–428. SAGE Publications Sage CA: Los Angeles, CA, 2003.

[49] Stefanos Nikolaidis, Przemyslaw Lasota, Ramya Ramakrishnan, and Julie Shah. Improved human–robot team performance through cross-training, an approach inspired by human team training practices. *The International Journal of Robotics Research*, 34(14):1711–1730, 2015.

[50] Julie Shah, James Wiken, Brian Williams, and Cynthia Breazeal. Improved human-robot team performance using chaski, a human-inspired plan execution system. In *Proceedings of the 6th international conference on Human-robot interaction*, pages 29–36. ACM, 2011.

[51] Jennifer L Burke, Robin R Murphy, Michael D Coovert, and Dawn L Riddle. Moonlight in miami: Field study of human-robot interaction in the context of an urban search and rescue disaster response training exercise. *Human–Computer Interaction*, 19(1-2):85–116, 2004.

[52] Brian G Bell and Nancy J Cooke. Cognitive ability correlates of performance on a team task. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 47, pages 1087–1091. SAGE Publications, 2003.

[53] PC Kyllonen, DJ Woltz, RE Christal, WC Tirre, VJ Shute, and S Chaiken. Cam-4: Computerized battery of cognitive ability tests. *Unpublished computer program, Brooks Air Force Base, TX*, 1990.

[54] PC Kyllonen. The cognitive abilities measurement (cam) battery version 4, reference manual. In *Technical Paper AL/HR-TP-0000*. Armstrong Laboratory Brooks AFB, TX, 1995.

[55] R Opperman. Adaptive user support. *Ergonomic Design of manually and Automatically Adaptable Software*, 1994.

[56] Reinhard Oppermann and R Rasher. Adaptability and adaptivity in learning systems. *Knowledge transfer*, 2:173–179, 1997.

[57] Christopher A Miller, Harry Funk, Robert Goldman, John Meisner, and Peggy Wu. Implications of adaptive vs. adaptable uis on decision making: Why "automated adaptiveness" is not always the right answer. In *Proceedings of the 1st international conference on augmented cognition*, pages 22–27, 2005.

[58] Esma Aïmeur, Claude Frasson, and Carmen Alexe. Towards new learning strategies in intelligent tutoring systems. In *Brazilian Symposium on Artificial Intelligence*, pages 121–130. Springer, 1995.

[59] Ewart de Visser, Brian Kidwell, John Payne, Li Lu, James Parker, Nathan Brooks, Timur Chabuk, Sarah Spriggs, Amos Freedy, Paul Scerri, et al. Best of both worlds design and evaluation of an adaptive delegation interface. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 57, pages 255–259. SAGE Publications, 2013.

[60] Jill L Drury, Justin Richer, Nathan Rackliffe, and Michael A Goodrich. Comparing situation awareness for two unmanned aerial vehicle human interface approaches. Technical report, DTIC Document, 2006.

[61] RM Taylor. Situational awareness rating technique(sart): The development of a tool for aircrew systems design. *AGARD, Situational Awareness in Aerospace Operations 17 p(SEE N 90-28972 23-53)*, 1990.

[62] Daniel Câmara. Cavalry to the rescue: Drones fleet to help rescuers operations over disasters scenarios. In *IEEE Conference on Antenna Measurements & Applications (CAMA)*, pages 1–4. IEEE, 2014.

[63] Kazi Tanvir Ahmed Siddiqui, David Feil-Seifer, Tianyi Jiang, Sonu Jose, Siming Liu, and Sushil Louis. Development of a swarm uav simulator integrating realistic motion control models for disaster operations. *arXiv preprint arXiv:1704.07335*, 2017.

[64] Ilan Kroo, Fritz Prinz, Michael Shantz, Peter Kunz, Gary Fay, Shelley Cheng, Tibor Fabian, and Chad Partridge. The mesicopter: A miniature rotorcraft concept–phase II interim report. *Stanford University*, 2000.

[65] John Grefenstette, Rajeev Gopal, Brian Rosmaita, and Dirk Van Gucht. Genetic algorithms for the traveling salesman problem. In *Proceedings of the first International Conference on Genetic Algorithms and their Applications*, pages 160–168. Lawrence Erlbaum, New Jersey (160-168), 1985.

[66] Paulo Zemek. Travelling salesman - genetic algorithm, 2015.

[67] Sandra G Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage Publications Sage CA: Los Angeles, CA, 2006.