STPA Hazard Analysis of Human Supervisory Control of Multiple Unmanned Aerials Systems

by

Elias B. Johnson

B.S. Mechanical Engineering
The United States Air Force Academy, 2017

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MAY 2021

Signature of Author:_____

Elias Johnson
Department of Aeronautics and Astronautics
May 18, 2021

Certified by:_____

Nancy Leveson
Professor, Aeronautics and Astronautics
Thesis Supervisor

Accepted by:_____

Zoltan Spakovszky
Professor, Aeronautics and Astronautics
Chair, Graduate Program Committee

Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

STPA Hazard Analysis of Human Supervisory Control of Multiple Unmanned Aerials Systems

by

Elias B. Johnson

Submitted to the Department of Aeronautics and Astronautics
on May 18, 2021 in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

ABSTRACT

Unmanned Aircraft Systems (UAS) operations are shifting from multiple operators controlling a single-UAS to a single operator supervising multiple-UAS engaged in complex mission sets. To enable this paradigm change, there is wide consensus in the literature that limitations in human cognitive capacity require shifting low-level control responsibilities to automation so that human operators can focus on supervisory control. However, hazard analyses to identify related safety concerns have largely used traditional hazard analysis techniques that cannot handle the level of complexity of these systems and none can provide recommendations for the early stages of system development. To begin to address this shortfall, this thesis applies System-Theoretic Process Analysis (STPA) on a model of a multi-UAS system with human-supervisory control. This hazard analysis approach handles complex software and human-machine control interactions together. This thesis details both how the hazard analysis was executed and the implications of the analysis results. Numerous traceable causal scenarios are systematically identified and used to generate design recommendations. These recommendations, if applied, will help ensure multi-UAS systems with human supervisory control are designed with safety in mind.

Thesis Supervisor: Nancy Leveson
Title: Professor of Aeronautics and Astronautics

[Page Intentionally Left Blank]

Acknowledgments

My experience at MIT has been truly exciting and memorable. Over the past two years, I have grown both professionally and personally thanks to several people. The following is an attempt to recognize the contribution of those who have helped me succeed while at MIT.

I would first like to thank my amazing wife, Inaya. You encouraged me to apply to MIT and uprooted your life when I was accepted. You have always supported me and that is especially true while here at MIT. I truly could not have completed this degree without your guidance, love, and support over these past two years. I know that together we can accomplish anything, and I am excited for all the journeys we get to explore together.

I am extremely thankful for my family and friends for both molding me into who I am today and providing support and encouragement while at MIT. Mom and Dad, thank you for all the phone calls, letters, care packages, and visits. You always know how to bring a little unexpected joy to my day and make me feel loved.

I owe an extreme debt of gratitude to my advisor Professor Nancy Leveson. Thank you for encouraging both my personal and academic development. You have opened my eyes to the importance of taking a *true* systems approach to the design of complex systems. I was incredibly fortunate to join the Systems Safety Research Group and I am excited to apply what you have taught me in my Air Force career and beyond.

I would also like to thank all the members of the System Safety Research Group that have been wonderful friends and assisted me in both classes and in writing this thesis. Dr. John Thomas thank you for your support and leadership within the group. Thank you to Andrew Kopeikin, Michael Schmidt, Lawrence Wong, Justin Poh, Adam Munekata, Dylan Muramoto, Sam Yoo, and Dro Gregorian.

I am also grateful to the United States Air Force for providing me this opportunity to attend MIT. I am truly humbled by the opportunities I have been provided so far in my career. I hope to use the knowledge I have gained in these past two years to make a meaningful contribution to our United States Air Force.

Table of Contents

Chapter 1: Introduction 11
1.1 Thesis Motivation 11
1.2 Thesis Scope 12
1.3 Thesis Objectives 13
1.4 Overview of Chapters 13
Chapter 2: Literature Review 14
2.1 Overview of Human-Supervisory Control of Multi-UAS Systems 14
2.2 Control Characteristics of Human Supervisory Control of Multi-UAS Systems 16
2.3 Review of Common Hazard Analysis Techniques 19
2.3.1 Preliminary Hazard Analysis 19
2.3.2 Fault Tree Analysis 21
2.3.3 Failure Mode and Effects Analysis 22
2.3.4 Systems-Theoretic Process Analysis 23
2.4 Review of Previous Multi-UAS Hazard Analyses 25
Chapter 3: Hazard Analysis of Human Supervisory Control of Multi-UAS 27
3.1 Hazard Analysis Overview 27
3.2 Defining the Losses, Hazards, and System Constraints 29
3.2.1 Overview of STPA Step One 29
3.2.2 Losses, Hazards, & Constraints for Human Supervisory Control of Multi-UAS 29
3.3 Developing the Hierarchal Control Model 32
3.3.1 Overview of STPA Step Two 32
3.2.2 Hierarchical Control Model: Human Supervisory Control of Multi-UAS 33
3.4 Identifying the Unsafe Control Actions 42
3.4.1 Overview of STPA Step Three 42
3.4.2 Unsafe Control Actions for Human Supervisory Control of Multi-UAS 42
3.4 Generating the Causal Loss Scenarios 44
3.4.1 Overview of STPA Step Four 44
3.4.2 Causal Loss Scenario for Human Supervisory Control of Multi-UAS 46
3.5 Summary of Hazard Analysis of Human-Supervisory Control of Multi-UAS 48
Chapter 4: Implications of STPA Hazard Analysis for the Human-Supervisory Control of Multi-UAS Systems 49

4.1 Overview of Chapter 49
4.1 Highlights of Recommendations Generated from Analysis 50
4.1.1 Recommendations to Improve Handoff between Multiple Controllers 50
4.1.2 Recommendations to Improve Feedback for Human-Machine Trust 51
4.1.3 Recommendations to Assist Operator in Providing Timely Plan Approval 52
4.1.4 Recommendations to Improve Operator to Machine Input Semantics 534
4.1.5 Recommendations to Improve UAS Task Assignments Generation 55
4.1 Summary of Design Recommendations Generated from Hazard Analysis 56
Chapter 5: Thesis Summary 57
5.1 Research Contributions 57
5.2 Future Work 57
Appendix A: STPA Unsafe Control Actions 62
Appendix B: Causal Loss Scenarios & Recommendations 69

List of Figures

Figure 1. Example Fault Tree Analysis 22
Figure 2. Generic Control Structure..... 32
Figure 3. Hierarchical Control Model of Human Supervisory Control of Multi-UAS 34
Figure 4. Breakdowns related Unsafe Controller Behavior and Unsafe Feedback Path .. 45
Figure 5. Breakdowns related to Unsafe Control Path and Unsafe Process Behavior 45

List of Tables

Table 1. Example PHA Table 19
Table 2. Example Failure Mode and Effects Criticality Analysis 21
Table 3. Mission Authority Control Actions 32
Table 4. Mission Authority Feedback 32
Table 5. Pre-Mission Planners Control Actions 33
Table 6. Pre-Mission Flight Planners Feedback 34
Table 7. UAS Operator Control Actions 39
Table 8. UAS Operator Feedback 39
Table 9. Multi-UAS Team Controller Control Actions 40
Table 10. Multi-UAS Team Controller Feedback 40
Table 11. Selected Operator Unsafe Control Actions 43
Table 12. Selected Recommendations related to Handoff between Multiple Controllers 50
Table 13. Selected Recommendations related to Human-Machine Trust Development .. 51
Table 14. Selected Recommendations related to Timely Plan Approval 52
Table 15. Selected Recommendations related to Human-Machine Input Semantics 53
Table 16. Selected Recommendations related to UAS Task Assignment Generation 54

[Page Intentionally Left Blank]

Chapter 1

Introduction

1.1 Thesis Motivation

This thesis has an extremely personal motivation. On June 15th, 2020 I received a call from my mother that an F-15C Strike Eagle from the 48th Fighter Wing at Royal Air Force Base Lakenheath had just crashed into the North Sea, killing the pilot. She hesitantly told me the pilot's name and asked me if I knew him. Upon hearing his name, I immediately fell to the ground in disbelief. The pilot was USAF 1Lt Kage Allen - a close friend. Kage and I both grew up in the same area of Utah and graduated from the United States Air Force Academy in 2017. During our time at the Academy, we had several classes together and bonded over late-night study sessions finishing up homework assignments due the next day. Kage was a beacon of positivity, a true friend, and will forever remain a personal hero to me.

Several months after Kage's crash, the accident investigation report was released. The report stated the accident was a result of "the *pilot's fixation* [emphasis added] on the intercept of the simulated adversary aircraft and failure to execute cockpit instrument visual scans" while flying through cloud cover and experiencing spatial disorientation [1]. Simply put, the accident investigation board concluded that Kage was unable to safely control his aircraft because he got overwhelmed and distracted by other tasks required of him during the mission. This type of report conclusion is not uncommon – all too often accidents are blamed on operators without considering how the system design contributed to the event [2]. I promised when I entered MIT, and reaffirmed after reading the report, that I would dedicate my life to ensuring that operators can effectively control the complex technical systems that we, as engineers, design.

Kage's accident is one of many examples that represents a much broader challenge in the development of complex, software-intensive control systems. Over the past 50 years, the flexibility of software has allowed engineers to develop control systems that can accomplish incredible feats. However, the development of software is both a blessing and a curse. The flexibility that software enables also allows for nearly unlimited system complexity that makes it difficult for designers to understand, predict, and mitigate against undesired system behaviors caused by non-obvious interactions among system components [2]. Unfortunately, most of the traditional hazard analysis techniques commonly practiced today are only able to identify component failures or oversimplify accidents that are caused by complex interactions between hardware, software, and humans. As a result, when these systems are fielded they may be unsafe because potential

hazards caused by these complex interactions were not properly evaluated and mitigated during the design process.

Fortunately, a relatively new hazard analysis method called Systems Theoretic Process Analysis was developed that can model and analyze complex system interactions [3]. The motivation for this thesis is that designing unsafe systems does not necessarily have to be the only way forward. If engineers can perform early hazard analysis of complex, software-intensive systems during development, there is an opportunity to mitigate safety concerns from the onset. The rest of this thesis is focused on a specific application of a hazard analysis performed early in system design for an emerging field – human supervisory control of multiple Unmanned Aerial Systems.

1.2 Thesis Scope

Unmanned Aerial Systems (UAS) developed and fielded in the 20th and early 21st century can be characterized by relatively simple mission sets where multiple operators were responsible for providing direct control of the flight, navigation, and payloads of a single system [4]. However, recent advancements in autonomous controller technology are shifting these responsibilities. Advanced autonomous controllers will be able to assume some of the low-level control responsibilities previously required by operators. This advancement allows fewer human resources per UAS, thereby increasing the feasibility of employing teams of multiple UAS that are capable of more complex missions. The role of the operator is expected to shift from providing direct control over a single UAS to a mission manager who supervises a team of UAS. There are now numerous civilian and government programs in development to operationalize this paradigm change.

However, there are still many challenges that must be addressed before multi-UAS systems become a reality. A major concern raised in the DoD FY 2017-2042 UAS Integrated Roadmap, and by others experts, is a lack of understanding of the complex interactions among the system components and their effects. This includes understanding interactions between operators and automation, and how they must work together to provide effective and safe control of multi-UAS systems in the more advanced and dynamic missions [5]. The human-supervisory control of multi-UAS systems represents a size shift in both mission complexity and control complexity. Therefore, it will be unsatisfactory to rely on previous UAS hazard analyses, design requirements, and operational insights to inform the development of these new systems.

To ensure operators and autonomous controllers can work together to safely control multiple UAS requires a rigorous early system hazard analysis to inform design requirements. Unfortunately, experts note that in general very few hazard analyzes of these systems have been performed [7]. In addition, the hazard analyzes that have been performed are (a) Preliminary Hazard Analyzes (PHAs), which provide limited design recommendations [6], or (b) examine specific UAS configurations and are not primarily focused on analyzing the human-supervisory control aspect of these systems [8], [9]. The gap in present hazard analyzes of these systems leads directly to the objectives of this thesis.

1.3 Thesis Objectives

The objective of this thesis is to demonstrate how safety-guided design recommendations can be developed that (1) enable safety to be designed into the system early when most effective, and (2) apply to a wide range of multi-UAS systems with human supervisory control.

> Research Question 1: How can a systems theoretic model of a multi-UAS system be created that is reflective of a broad range of multi-UAS systems with human supervisory control?
>
> Research Questions 2: How can a hazard analysis be performed to provide insights for the early phases of concept development of multi-UAS systems with human-supervisory control?

Research question 1) is answered with the development of an abstracted model of a multi-UAS system with human supervisory control in Chapter 3. Research question 2) is answered by an STPA hazard analysis performed on such an abstracted model and the resulting recommendations provided in Chapter 4.

1.4 Overview of Chapters

Chapter 2 presents a literature review of multi-UAS systems, hazard analysis methods, and previous work related to this thesis. Chapter 3 presents the modeling and hazard analysis of a multi-UAS system with human-supervisory control. Chapter 4 presents an overview of the design recommendations that were systematically generated from the STPA hazard analysis. Chapter 5 provides a conclusion of the thesis, along with areas for further development.

Chapter 2

Literature Review

The purpose of this literature review is to provide a review of multi-UAS systems, hazard analysis methods, and previous work that lay the foundation for this thesis. This literature review is organized into four sections. The first section provides an overview of human-supervisory control of multiple UAS systems. The second section describes the expected control characteristics of multi-UAS systems that serve as a foundation to develop a control model of these systems for the STPA hazard analysis. The third section presents a summary and critique of common hazard analysis techniques and explains why Systems Theoretic Process Analysis was chosen in this thesis. The final section explores previous hazard analyses related to human-supervisory control of multiple UAS systems and explains why additional research is required.

2.1 Overview of Human-Supervisory Control of Multi-UAS Systems

There have been a wide range of applications proposed for multi-UAS systems in both the civilian and military sectors. Some of the applications include home goods delivery [10], surveying and mapping [11], search and rescue [12], disaster response [13], and numerous others [14]–[18]. The implementations of these applications vary in size, configuration, and function of the UAS required for the mission. However, a common characteristic across all applications is the coordinated control of multiple UAS to achieve a synergistic effect that would not be possible with a single UAS. A recent survey of 65 proposed multi-UAS projects indicated that there are four primary perceived advantages to multi-UAS systems when compared to single UAS systems [7]:

(1) Time Efficiency – The employment of multiple UAS systems can significantly reduce the operational time required to complete a mission. The most drastic difference in operational length between single and multiple UAS systems can be found in missions that require the collection of information over an expansive area. This advantage becomes particularly important in time-sensitive emergencies like the detection of nuclear radiation after a disaster, or search and rescue operations.

(2) Reduced Cost – The distribution of mission capabilities and payloads across multiple UAS can significantly reduce the size and complexity previously required in a single UAS system. The reduction in size and complexity of each UAS has been shown to reduce the overall system cost.

(3) - Fault Tolerance in the Event of a Loss – In single UAS systems, the mission hinges on the ability of a single vehicle to complete the objectives. In the event of a component or system failure, the ability to complete the mission objectives may be completely lost. However, in multi-UAS systems, it may be possible to mitigate the loss of a single vehicle by redistributing the tasks of the lost UAS to other capable UAS team members. This advantage, of course, is only applicable when the failure is constrained to a single UAS and not the entire team.

(4) – Increased Mission Flexibility – Multiple UAS systems allow for increased mission flexibility that may be required for dynamic operational environments. Single UAS systems can only perform one task at a time. However, a team of UAS can be dynamically allocated to perform multiple tasks at the same time and rearranged as necessary. One example is a mission where a team of UAS is required to perform target tracking on a large group of people. If at some point, a portion of the group separates from the main group, tasks could be distributed amongst the UAS to perform individual target monitoring.

To enable the benefits of multi-UAS operations requires an operational shift from multiple operators remotely *controlling* a single-UAS to a single operator *supervising* multiple-UAS [6]. In this context, the difference between operator *control* and *supervision* is characterized by a shift in the delineation of control responsibilities between the operator and the UAS autonomous controllers [19]. Operators that control UAS are responsible for providing lower-level control inputs directly to UAS flight, navigation, and payload sub-systems to achieve the flight and mission objectives. In contrast, when operators perform supervision of UAS the responsibility for lower-level control and UAS team coordination is delegated to automated controllers [4]. The primary responsibility of the operators in supervision is to provide higher-level control inputs related to the overall coordination of the mission. In several examples of supervisory control in multi-UAS implementations, the operator will provide mission planning parameters to an autonomous controller so that it can develop plans and present them to the operator for review [4].

The allocation of more control responsibilities to automated controllers has the potential to increase the mission reach without increasing human operator resource requirements. For example, early studies showed that a single operator could only control 4-5 vehicles, but they could supervise around 12 UAS at a time [20]. However, increasing use of automation also introduces new safety and human factors concerns which have been raised extensively in the literature [6]. For example, the skills and training required for operators to perform supervisory control may be considerably different than those previously required in lower-level control. Furthermore, in certain conditions, the UAS operator may have to override the automation and revert to lower-

level control, potentially leading to cognitive overload if the system is not designed to account for these situations [2].

Significant effort has been made to develop advanced autonomous controllers that appropriately assist operators in coordinating the tasks of multiple UAS during the mission [21]. However, even with considerable advancements, there is a wide consensus among experts that multi-UAS systems are unlikely to be fully autonomous in the near future. This is largely driven by limitations in autonomous controller technology [21], legal restrictions [22], and increased operational risk without any operator oversight [6]. As a result, there is an acceptance that both humans and automation will be required to work together to provide effective and safe coordination and control over multiple UAS both before and during a mission.

2.2 Control Characteristics for the Modeling of Human Supervisory Control of Multi-UAS Systems

The purpose of the following sub-section is to provide a summary of the expected control characteristics of future multi-UAS systems. This information provides a foundation to develop an abstracted control model of these systems for the hazard analysis presented in Chapter 3. The review includes an examination of control characteristics both before and during flight distributed between humans and UAS autonomous controllers.

In many multi-UAS applications, the operators may receive guidance on the objectives and restrictions of the mission from a higher mission authority [17]. Guidance may be developed and sent before the mission so pre-mission activities can be accomplished. For example, the guidance may be used to determine the required number of UAS or the expected mission tasks. Guidance may also be updated during the mission to ensure operators have an up-to-date understanding of what needs to be accomplished. For example, data collection priorities may change during the mission, or the operators may need additional permissions to fly over new air spaces.

Before UAS flight operations, there is expected to be a certain amount of pre-mission planning provided by the operator to configure the UAS team. In pre-planning, users may specify parameters to control the system during flight. Pre-mission inputs may include pre-programmed waypoints for each UAS, no-fly zones, or a list of tasks and their priorities for different phases of the mission. This also includes updates to the Team software required to allocate tasks to each UAS during the mission [23].

An advantage of pre-mission input and plan development is that it can reduce operator workload during flight operations. It may also allow for faster mission execution once the UAS team has entered the airspace - which can be critically important for time-sensitive applications. Although pre-mission plan development is important, it is not expected to be the means of control for the entire mission. Full reliance on pre-mission plan development does not allow enough system flexibility to account for changes during complex and dynamic mission environments. Therefore, almost every implementation of future multi-UAS systems represented in the literature also involves real-time operator and automation interaction during flight [7].

During flight operations, it is expected that the operator and autonomous controllers will work together to provide control over the UAS. Several human factors studies indicate that the responsibility to coordinate the task assignments of more than a few UAS during flight would exceed the cognitive abilities of human operators [24]. As a result, the coordination and plan development for multi-UAS systems during flight is delegated to an autonomous controller. The operator is responsible for providing plan input parameters related to the mission. For example, in one reconnaissance application, operators were responsible for defining and updating a desired target search area throughout the mission [25]. These parameters may vary depending on the mission application. These inputs can be provided throughout the mission to account for changes in the objectives or environment.

A key control characteristic of future multiple UAS systems is the responsibility of an autonomous controller to develop plans and allocate UAS resources based on the operator inputs. The literature shows that plan development could be accomplished with fixed algorithm assignment or dynamic algorithm assignment [7]. In a fixed algorithmic assignment, the autonomous controller develops plans for each UAS several stages in advance. This could be the case, for example, in the coordination of multiple flying digital displays where each UAS is provided a route for the entire mission at the beginning of flight operations [26]. However, this type of planning will only apply to very simple mission sets. Therefore, the most common type of plan development is dynamic algorithm assignment where the autonomous controller actively analyzes information obtained from each UAS and other sources and decides on how to best assign UAS resources on a continuous cycle [27].

There are also different mechanisms to develop plans and systematically assign tasks to each UAS during the mission. This control can range on a spectrum from fully centralized to fully decentralized control. In centralized control, a single autonomous controller provides commands to each induvial UAS. This is representative of how the Intel UAS light show demonstration in 2016 was implemented [28]. In decentralized control, no single controller oversees the entire plan development. Instead, decisions are made by each UAS and usually involve establishing shared consensus throughout the team. A common way this has been proposed is a bidding system where each UAS bids on tasks it believes it should accomplish, and tasks are assigned based on the bids of each UAS [29]. The use of either centralized or decentralized control networks has advantages and disadvantages. The important realization is that different systems may employ different control structures. Therefore, a broad analysis of these systems should consider the full range of implementations options.

The control interactions between operators and autonomous controllers continue from plan development into plan approval and execution. In some UAS implementations, the plans developed by the autonomous team controller must be approved by the operator before execution [30]. In other implementations, more control responsibility is given to the autonomous controller to execute plans without operator consent. In this case, the operator only provides intervention when they believe an improper plan or task assignment has been developed [31].

There has been a considerable effort in the human factors research community to define the appropriate level of delegation of control between humans and UAS automation for plan approval and execution [32]–[34]. Overall, the research on the safety

and effectiveness of the two options is mixed and is largely dependent on many factors including operator workload, reliability of the automation, and the complexity of the task or mission [19]. Some studies also indicate it may be more effective to have a fluid level of control responsibility between the operator and UAS that dynamically changes throughout the mission based on the perceived risk of the task [35], [36]. For instance, the operator may allow the UAS to proceed without consent for some tasks, but require consent for other tasks which if improperly executed could lead to irreparable harm. A specific example for a military application may be the difference in required approval between plans developed for a reconnaissance phase of a mission versus those for target acquisition and weapons selection for targets identified during the reconnaissance [37]. The important realization is that systems may involve varying responsibilities required of the operator to review the plans developed by the autonomous team controller. A broad analysis of these systems should consider a range of implementations options.

Most of the preceding discussion has focused on the coordination of multiple UAS to achieve the mission objectives. However, the control of each UAS is also important to consider. The literature reflects that onboard each UAS there will be an auto-pilot that controls the flight and navigation of each system [38]. The auto-pilot may manipulate the flight and payloads of the physical UAS to achieve the tasks assigned to the UAS. The operator may have the responsibility to step in if the autopilot is unable to adequately control the UAS. However, this type of low-level control would severely limit the pilot's ability to manage other UAS as well as other aspects of the mission and is only expected under emergency circumstances.

In summary, this section of the review presented an overview of the expected control characteristics for human-supervisory control of multi-UAS systems. It reflects that humans are expected to provide both input planning parameters before and during flight. An autonomous controller uses these inputs and feedback from the UAS and environment to dynamically develop plans for each UAS to accomplish the mission objectives. Human operators provide oversight of the plan development and execution throughout the mission. This review provides a foundation to develop an abstracted control model of these systems required for the hazard analysis presented in Chapter Three.

2.3 Review of Common Hazard Analysis Techniques

In this section, several of the most common hazard analysis techniques are examined. Each of the first three sub-sections provides an overview of a traditional hazard analysis method and explores why it is insufficient for the analysis of human-supervisory control of multi-UAS systems. The final sub-section provides an overview of STPA and a justification for the selection of this method in the analysis presented in this thesis.

2.3.1 Preliminary Hazard Analysis

Preliminary Hazard Analysis (PHA) is a method for the identification of hazards during the early stage in the design process. The Preliminary Hazard Analysis is used to obtain an initial risk assessment of the system hazards usually before detailed design information is available. The analysis is based on available data including accident information from similar systems and other lessons learned. The content of a PHA typically includes (1) a hazardous condition (2) potential causes of the hazardous condition (3) major effects of the hazardous event, (4) assessment of the probability and severity level of the effect, and (5) potential preventative measures. The analysis results are normally captured in a table. An example PHA pulled from the literature for a nuclear reactor is presented in Table 1 [39].

Table 1. Example PHA Table

HAZARDOUS CONDITION	**CAUSE**	**EFFECT**	**ASSESSMENT**	**CORRECTIVE MEASURE**
Damage to feed reactor tube	Feed compressor failure (no endothermic reactions in reactor)	Capital loss, down time	Low probability, medium consequence	Provide spare compressor with automatic switch-off control
Explosion, fire	Pressure build-up in the reactor due to plug in transfer lines	Fatalities, injuries	Low probability, high consequence	Provide pressure relief valve on reactor tubes

The main advantage of PHA is that it can be applied before details of the system have been determined. However, this lack of specificity also limits its applicability and effectiveness. The types of causes identified by PHA techniques are often very generic. For example, a recent PHA in the aerospace domain listed “design flaws, coding error, software operating system problems” and “human error” as potential hazard causes [40]. These generic types of causes are not particularly useful for developing detailed requirements for a system.

An important note, and an additional limitation, is that the hazardous conditions identified in a PHA are not hazards. A hazard defines the system conditions and environmental factors that may lead to a mishap or undesirable system loss [2]. The hazardous conditions identified in a PHA are more indicative of a mishap or undesirable loss event. For example, in Table 1 the hazardous condition "explosion, fire" is an undesirable accident or loss that must be prevented. An appropriate hazard would define the nuclear reactor system conditions and environmental factors that may lead to an explosion or fire.

Another limitation of PHA is that it generally relies on previous accident investigation results or operational assessments to determine the initial hazardous condition [6]. Prior knowledge of system hazardous conditions is unlikely to exist for paradigm-changing systems like future multiple UAS systems with human-supervisory control. Because of the limitations discussed above, PHA was not selected as an appropriate hazard analysis tool for this thesis.

2.3.3 Failure, Mode, Effects, and Criticality Analysis

While reliability and safety are very different qualities of a system, the two can be interrelated. Accidents can occur because of the failure of unreliable system components that affect the safety of the system. However, it is important to note that not every accident is caused by the failure of unreliable components. A Failure Modes, Effects, and Criticality Analysis (FMECA) is a method to assess the reliability of a system based on its subsystems and components and develop possible mitigations to improve system reliability. FMECA is a bottom-up approach that examines each component in isolation and then determines the effect on the overall system.

The first step of FMECA is to identify and list all of the system components. In large complex systems, this list may be very large. The next step is to determine all of the possible failure modes of each component by considering all possible operating modes of the component. For each failure mode, the probability of occurrence and the potential causes for the failure mode are then generated. The probability of each failure mode is predicted from manufactures testing data or operational experience based on a specific environmental setting.

The next step of FMECA is to determine the effect and criticality of each failure mode on all other components and the overall system. In general, the criticality is divided into those that result in system failure, labeled critical, and those that do not, labeled non-critical. An important note here is that not every component failure mode will have an unsafe effect on the entire system. It is possible to generate lists of possible failure modes that have no result on the safety of the overall system [41].

From the results of the FMECA, possible mitigations can be developed to improve the reliability of the system. Table 2 shows an example FMECA for a railroad crossing boom gate. This gate is weighted in such a way that the weight of the boom gate will lower the gate in the event of a power failure [8].

Table 2. Example Failure Mode, and Effects Criticality Analysis

Component	Failure Modes	Cause(s) of failure	Probability	Effects	Criticality Level	Possible Mitigations
Railroad Crossing Bottom Gate	Up	Frozen component, obstruction	0.001	Cars on the track when a train is approach -ing	Severe	Install system indicating boom failed to lower completely
	Down	Motor failure, loss of power, obstruction	0.005	Cars cannot enter train crossing	Minimal	Acceptable as is

One strength of FMECA is the thoroughness required in the evaluation, at least in terms of failures. Each component and the possible failure modes are examined and recommendations can be developed to improve the overall system reliability. However, the bottom-up approach also has its drawbacks. Because FMECA begins the analysis at the component level it can only be performed late in system development. This is problematic because once the detailed components have been determined it may be both cost and schedule prohibitive to make changes or recommendations to affect the design of the system. The bottom-up approach is also time-consuming as each component must be examined and it is only determined later in the analysis if the failure will result in an undesirable system state.

Another limitation of FMECA is that when used as a hazard analysis tool, rather than just a reliability tool, it only captures a subset of potential hazards and accidents – those caused by component failure. FMECA was not created to capture accidents caused by unsafe interactions among components that are not the result of failure [41]. Most accidents in complex, software-intensive systems are caused by unsafe interactions that are not related to component failures. Considering the limitation discussed above, FMECA is not an appropriate hazard analysis to perform early in the development of future multi-UAS systems with complex interactions expected between humans, software, and hardware components.

2.3.2 Fault Tree Analysis

A common hazard analysis technique used in the aerospace, electronics, and nuclear industry later in system development is Fault Tree Analysis (FTA). The method is based on a chain of events model where accidents are assumed to occur because of a certain sequence of events or component failures that lead to unsafe system behavior.

The analysis begins with the definition of a set of system-level, undesired events. This is a critically important step because the rest of the analysis determines the specific ways in which this hazardous event may occur. After the system-level, undesired events are determined, the causal events related to the top event are traced into event or component failure branches at lower levels of the diagram. Each branch uses Boolean logic to describe the relationship between the events. After the branches have been

specified, the minimum combination of events that will lead to the system-level hazard is determined based on the Boolean logic in the diagram. It is also common to assign a probability to each event and combine the results to determine the probability of the entire sequence occurring. An example of a Fault Tree Analysis selected from the literature is shown in Figure 1 for a fire protection system [41]. The system-level hazardous condition is specified as "failure of the protection system" and the branches below detail the possible ways the system-level event may occur.

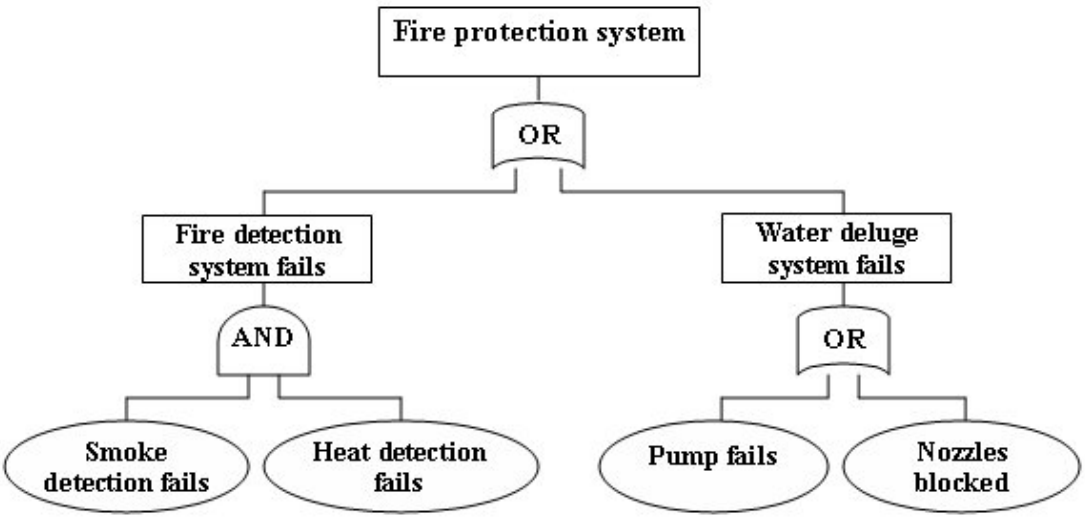

Figure 1. Example Fault Tree Analysis [41]

One strength of FTA is that event and component failures can be connected directly to the system-level hazardous event. This is accomplished by tracing the event sequence or component failures up through the branches of the diagram. The analysis results can be used to identify areas where system redundancy may be required to prevent a hazard in the event of a component failure or stop a sequence of events from occurring. However, there are several limitations of this method.

One limitation of FTA is that it assumes independence between each set of events or component failures in the diagram. For example, in the diagram above the failure of the fire detection system is considered separately from the failure of the fire suppression system. In complex, software-intensive systems the events and system components may occur or interact in unknown or unforeseen ways and the independence assumption between single events may not always be provable or true. Even in cases where the events are independent, estimating the expected probability of a set of events is based on other assumptions that may be inaccurate, thus resulting in unrealistic probability assessment [42].

Because both FTA and FMCEA assume accidents occur because of a sequence of events or component failures they have overlapping limitations. Similar to FMCEA, FTA only captures accidents caused by component failures and omits accidents caused by non-component failures, including system design errors or missing requirements. In addition, because the development of the fault tree requires detailed knowledge of the system components and their knowledge which is not available until after the system has been

developed. As a result, the ability to provide detailed design recommendations early in system development is limited [41].

Another limitation of FTA is that it assumes independence between each set of events or component failures in the diagram. For example, in the diagram above the failure of the fire detection system is considered separately from the failure of the fire suppression system. In complex, software-intensive systems the events and system components may occur or interact in unknown or unforeseen ways and the independence assumption between single events may not always be provable or true. Even in cases where the events are independent, estimating the expected probability of a set of events is based on other assumptions that may be inaccurate, thus resulting in unrealistic probability assessment [41]. An additional limitation of FTA is the oversimplification of hazardous events into a chain of failure event models using Boolean logic. The structure of the fault tree analysis captures discrete failure events that occur at a given period in time. However, it does not convey information about time- and rate-dependent events, degrees of partial failure, and dynamic system behavior. This over-simplification is important in systems where there may be multiple phases of a mission where the system passes through more than one phase of operation. As a result, accidents that may occur during or because of transition from one system state to another are not captured [42].

Multi-UAS applications are expected to be dynamic processes where the system transitions between multiple states and the mission may occur over several phases. FTA would be unable to appropriately identify hazards related to this characteristic of these systems. In addition, multi-UAS operations will involve complex interactions between operators and automation that may not interact in independent and obvious ways. So, the independence assumption required for FTA will not hold. For these reasons, FTA would not be a suitable hazard analysis technique for use in this thesis.

2.3.4 Systems-Theoretic Accident Model and Processes & Systems-Theoretic Process Analysis

Systems-Theoretic Accident Model and Processes (STAMP) is the name of a relatively new accident causality model based on systems theory. STAMP expands the traditional model of causality beyond a chain of directly related failure events or just component failures to include more complex processes and unsafe interactions among system components. In STAMP, safety is treated as a dynamic control problem rather than a failure prevention problem. The safety of a system is considered an emergent property. In other words, safety can only be considered when examining the complex interactions among system components, and not through a separate analysis of each component. Accidents occur because of the failure of controllers to enforce required safety constraints on controlled processes. Systems Theoretic Process Analysis (STPA) is a top-down hazard analysis tool based on the STAMP model that safety (and other emergent properties) can be treated as a dynamic controls problem [2].

The first step in an STPA hazard analysis is to define the undesirable losses that must be mitigated through the design and operation of the system. The losses can include traditional hazard analysis goals, but can also include broader categories related to security, privacy system performance, or other emergent properties. Hazards are then

defined as the system states or conditions that may lead to an accident in a worst-case scenario.

The second step is to build a model of the system called a control structure. A control structure captures functional relationships and interactions by modeling the system as a set of feedback control loops. In general, a controller provides control actions to a controlled process and receives feedback. The system model can include complex interactions among hardware, software, and human operators.

The next two steps of STPA analyze both when and why unsafe interactions may occur in the control structure. In the third step, the conditions under which the control actions may lead to system-level hazard, and ultimately a loss, are identified. These unsafe control actions are used to create functional constraints on the system to control safety as an emergent property. The fourth step identifies causal loss scenarios which examine reasons why each unsafe control action might occur in the system. These causal loss scenarios explain how breakdowns in the feedback control structure caused by incorrect feedback, improper control execution, inadequate requirements, component failures, and other factors could ultimately lead to losses.

After the causal loss scenarios are developed for each unsafe control, the information can be used to create requirements, identify mitigations, provide inputs to the design of the system architecture, evaluate/revisit existing design decisions and identify gaps in the design, define, test cases and create test plans, and for other uses. STPA enables analysis of complex systems that are controlled by both software and human operators which is a key characteristic of future multi-UAS systems. It is effective throughout the systems engineering process, especially in the early stages of system development where changes to the design can be made based on the analysis results. For these reasons, it is an ideal method for analyzing the complex interactions among future multi-UAS systems with human supervisory control [3].

2.4 Review of Previous Multi-UAS Hazard Analyses

Given the importance of designing safety into future multi-UAS systems with human-supervisory control, experts note that there has been a surprising lack of hazard analyzes of these systems available in the public domain [6], [7]. The author identified several hazard analyzes related to UAS operations and a handful related to multi-UAS operations. The following section provides a review of each and discusses why additional hazard analysis efforts are required.

The first was a hazard identification and analysis of small unmanned aerial systems using a PHA [6]. Belcastro et al. compiled civilian accident and incident investigation reports of 100 small UAS and determined common causal factors in the accidents. Themes from each accident were compiled and analyzed to identify current hazards and causes of hazards for small UAS systems. This information was used to develop a list of future potential hazards, causes of hazards, and safety risks anticipated with more complex future multi-UAS applications. These future potential hazards include general categories such as aircraft loss of control, failure of communication link, failure to avoid collision with terrain or moving obstacles, and others. However, the authors note that future multi-UAS applications may introduce safety risks that cannot be revealed by solely analyzing current and past mishaps. The results of the PHA work do not provide design recommendations, rather only broad categories of hazards. So, the helpfulness in the early design process is limited, and additional hazard analysis efforts are required.

The second is a master's thesis by Folse that applied STPA to analyze small unmanned aerial systems at Edwards Air Force Base for test and evaluation operations [8]. The hazard analysis primarily focused on analyzing interactions between UAS operators, air traffic controllers, and other test support personnel. The analysis systematically developed design recommendations to safely integrate small UAS into the congested airspace at Edwards Air Force Base. However, it is important to note that the focus was not on analyzing interactions between operators and advanced UAS automation in the control of multi-UAS systems. The work provides a foundation in the development of requirements for UAS control in general, but additional analysis is required to develop specific design recommendations for human-supervisory control of multi-UAS systems.

The third is a master's thesis that demonstrated how STPA could be incorporated in the U.S. Air Force Acquisition Process [43]. As part of the thesis, Summers performed STPA on a single UAS system. Consistent with the current implementation of single UAS systems, the operator in command provides direct control over the flight, navigation, and payloads of the vehicle. Summers derived unsafe control actions for the system but did not generate detailed casual loss scenarios. An important note is that she did not evaluate multi-UAS systems as it was not the primary focus of the analysis. However, her analysis results may be applicable when multi-UAS operators are required to revert to lower-level control inputs in emergencies.

The fourth is a master's thesis that demonstrated how STPA can be used to perform system architecture trade studies during early concept development [44]. As an example of the trade study process, Horney compared two potential control designs of a tethered UAS system. In the thesis, the controller that implements that command to set a formation shape for the team of UAS was evaluated. In one system design, the human

pilot in command was responsible for selecting a formation shape for the UAS. In the alternative system design, an autonomous controller was responsible for selecting the formation shape based on the present conditions and the current phase of flight. In the second design, the operator was responsible for intervening if the system could not select an appropriate formation shape. The analysis compared the hazard analysis results of each system design and systematically generated design recommendations for each. The hazard analysis of each system design generated for the tethered multi-UAS system provides a foundation for additional hazard analyzes that may analyze other aspects of human-supervisory control of multi-UAS systems.

The fifth is a master's thesis by Johnson that applied STPA to a manned-unmanned UAV team conducting a reconnaissance and target acquisition mission [9]. The focus of the thesis was analyzing interactions between a UAS operator, an autonomous loyal wingman, and other support personnel. The analysis systematically developed system requirements for safe operations of a loyal wingman UAS system. An important note is that the modeling of the system focused on a loyal wingman multi-UAS systems with a specific application. Additional hazard analyzes could build upon this work by focusing on other human-supervisory control aspects and applications of multi-UAS systems.

Overall, the previous analyzes have performed initial hazard identification using PHA, examined single unmanned aerial systems, and focused on specific characteristics or implementation of multi-UAS systems. Additional hazard analysis could expand upon the previous work by analyzing other human-supervisory control characteristics of multi-UAS systems identified in the literature review. This extension could include analysis of multi-UAS systems where an operator works with an autonomous controller to develop, review and execute plans for a team of UAS. The rest of this thesis focuses on performing a hazard analysis on such a system to inform potential design requirements.

Chapter 3

Hazard Analysis of Human Supervisory Control of Multi-UAS Systems

3.1 Hazard Analysis Overview

The purpose of this chapter is to present an STPA hazard analysis for the human supervisory control of multi-UAS systems. This chapter is organized into five sections. The first four sections correspond to the four steps of STPA, and the final section provides an overview of the analysis. In the first section, the unacceptable losses that must be avoided during operations are identified along with the hazardous system states that may lead to the identified losses. System safety constraints are established that must be enforced to ensure the unacceptable losses do not occur during operations.

In the second section, an abstracted, hierarchical control model of a multi-UAS system with human-supervisory control is presented that is representative of a wide range of multi—UAS applications or implementations. This model is an abstraction of multi-UAS systems which emphasizes how components interact using feedback control loops.

In the third section, interactions among the system components in the control model are examined to determine when control actions could lead to unacceptable losses. Several examples of unsafe control actions are provided which are reflective of the entire analysis.

In the fourth section, several examples of the loss scenarios generated by the full analysis are presented. These scenarios describe the causal factors that explain why an unsafe control action may occur. The results of the hazard analysis presented in this chapter are then used to develop recommendations for the design and operation of future multi-UAS systems with human supervisory control in the following chapter.

Before the results of the hazard analysis are presented it is important to establish what was included as part of the system boundary, and why it was selected. Achieving the desired applications of future multi-UAS systems requires both humans and autonomous controllers to work together to provide effective and safe control. The literature review reflects there will likely be multiple personnel—both before and during the mission—who are responsible for configuring both the autonomous controllers and the individual UAS and overseeing UAS team plan development, approval, and execution. The use of autonomous controllers will include both autopilot functions to

control the flight, navigation, and payloads of each UAS, as well as autonomous team controllers that optimize and dynamically develop plans for the UAS as a team.

The definition of the system boundary for this analysis includes the UAS, automation to control the UAS, personnel controlling the UAS during operations, and others who influence the system, such as those who configure the system before the mission or provide guidance on the objectives of the mission. Including both humans and non-human aspects within the system boundary establishes a holistic approach that allows the analysis, and any subsequent design recommendations, to be made based on the actual future operations of the system.

One challenge with any hazard analysis is managing the system complexity. The analyst must balance the usefulness of the analysis results on one end with the difficulty of examining the system in its entirety on the other. In this thesis, the analysis of multi-UAS systems with human supervisory control is managed with a systems theoretic approach based on functional abstraction. In this approach, certain details within the system boundary are abstracted without diminishing important interactions among the functional components of the system. This approach allows for emergent properties caused by interactions among the functional components to be analyzed and controlled before details and complexity are added by refining the system into sub-components. The functional abstraction focuses on the function of the components provided rather than the specific details of how the functional components are implemented.

Based on the results of the literature review, several functions are identified in this thesis that are important to consider for a hazard analysis that is focused on the human supervisory control of multi-UAS systems. These functions have been abstracted and allocated to five interacting controllers and controlled processes based on expected implementations. The Mission Authority controller is a functional abstraction of the entities who provide higher-level guidance on the objectives of the mission to those controlling the UAS. The Pre-Mission Planner controller is a functional abstraction of the entities who provide pre-mission plan management and configure the UAS before the mission so they can be operated as a team. The Operator controller is a functional abstraction of the human(s) who provides human-supervisory control of the UAS during operations. The Multi-UAS Team controller is the functional abstraction of any automated controllers that coordinate the actions of multiple UAS to achieve the mission objectives. The UAS(s) controlled process is the functional abstraction of all the UAS that are a part of the team.

Functional abstraction was used to manage the complexity of each one of the components inside the system boundary discussed above. For example, the Pre-Mission Planners controller may include personnel spread out over multiple geographical locations or reside in different organizations. From the literature review, the Multi-UAS Team Controller could be implemented as a centralized or decentralized network, However, in this analysis, the specific details of how each function could be implemented are not considered. Rather the emphasis is on understanding how each of the functional components interacts to achieve the mission. The interactions and responsibilities of each of the functional components identified above will be systematically determined during the development of the hierarchical control model.

3.2 Defining the Losses, Hazards, and System Constraints

3.2.1 Overview of STPA Step One

The first step of STPA is to define the purpose of the analysis. This begins by identifying the unacceptable losses that must be avoided during operations. The definition of a loss in STPA is broad and can involve anything of value to the stakeholders [3]. These losses can include losses that are traditionally considered in safety hazard analyses like loss of human life or injury. However, in STPA losses can also include a broader category of undesirable events including the loss or inability to perform a mission, the loss of company reputation, or the loss of critically protected information.

Once the unacceptable losses are identified, system hazards are derived. In STPA a distinction is made between the losses and system hazards. The losses include aspects of the environment over which operators and designers may have limited or no control. In contrast, the system-level hazards represent the system states that designers and operators have more influence to control. System hazards consist of a "set of conditions that, together with a particular set of worst-case environmental concerns, will lead to a mishap" [3]. The system hazards are then translated into constraints that specify the conditions or behavior of the system that must be satisfied to ensure the hazards do not occur.

3.2.2 Losses, Hazards, and System Constraints for Human Supervisory Control of Multi-UAS Systems

Three unacceptable losses that must not occur during operations were identified as the focus of this hazard analysis. These losses include not achieving both safe and effective operations. Each of the losses below is denoted by an identifier (L-) followed by a number for traceability throughout the rest of the analysis. This traceability ensures that each subsequent step in the analysis, and ultimately the design recommendations, are connected to the undesirable system losses that must be avoided.

L-1: Loss of life or Injury to People

Multi-UAS operations must not lead to the physical harm of any person that may interact with the system in its environment.

L-2: Loss or damage to UAS or other equipment in the environment

System operations must not lead to damage to the UAS or unintended damage to equipment in the UAS environment. If this loss occurs it could increase the cost of operations thereby reducing mission effectiveness and negatively impacting the perception of the safety of multi-UAS systems.

L-3: Loss of ability to complete the mission

The system must be able to accomplish the established mission objectives. For this analysis, the loss of mission is kept broad to encapsulate numerous multi-UAS applications. For example, it encapsulates the "loss of ability to provide

aerial reconnaissance" for a military application, or "loss of ability to deliver packages to the customer on time" in a home goods delivery application.

The losses identified in this thesis represent some of the most critical aspects of multi-UAS operations that must not be violated. A distinction is now made between these losses and potential system hazards. The losses identified above include aspects of the environment over which operators and designers of multi-UAS systems may have limited or no control. For example, other aircraft may enter the airspace designated for the multi-UAS system. The entry of another aircraft into UAS airspace is largely outside of the control of the system designers. However, the designers do have the ability to control the system state or response when this situation occurs. The designers can include the ability of the operators to choose whether to (1) take evasive action to avoid the other aircraft, or (2) do nothing to avoid the other aircraft. In the latter, a collision may occur because of a combination of the environmental condition, another aircraft entering the air space, and the state of the system in not avoiding the other aircraft. Three hazardous system states that could lead to the undesirable losses were identified for this analysis, and are denoted by (H-) for traceability throughout the analysis.

H-1: UAS is uncontrollable (either by the operator or automated controller) [L-1, L-2, L-3]

A concern with the operation of multi-UAS systems is the loss of vehicle control. If the UAS is unable to be controlled it may lead to loss of life, damage to UAS or other equipment, or loss of mission.

H-2: UAS violates minimum separation requirements between it and objects in the environment [L-1, L-2, L-3]

In multi-UAS applications, the UAS may be employed in congested air spaces and near objects in the environment. If the multi-UAS system violates the minimum separation requirements between itself and other objects in the environment this could lead to loss of life or injury to people; loss or damage to UAS or other equipment in the environment; or a loss of ability to complete the mission.

H-3: The system is unable to complete the mission objectives [L-3]

This hazard is abstracted to encapsulate any of the system states that would prevent the system from completing the mission. For example, for a military UAS, the hazard may be considered as "the system is unable to collect reconnaissance data", or "the system is unable to engage the required target".

These hazards must be avoided during the operation of future multi-UAS systems. To ensure these hazards do not occur requires constraints to be imposed upon the system. These constraints are the conditions or behaviors of the system that need to be satisfied to prevent the hazard. From the hazards, three system constraints were identified in this analysis. The focus of the rest of the analysis is determining when and

why these constraints may not be adequately enforced in the system. Each system constraint is denoted by (SC-) for traceability throughout the analysis.

SC-1: The system must be controllable by the operator, or automated controller during the mission [H-1].

This system constraint is the reciprocal of H-1, and it must be enforced to prevent the potential loss of life or injury, loss or damage to the UAS or other equipment in the environment, and loss of ability to complete the mission.

SC-2: The UAS must satisfy the minimum separation requirements from objects in the environment or other UAS [H-2].

This system constraint is the reciprocal of H-2, and it must be enforced to prevent the potential loss of life or injury, loss or damage to the UAS or other equipment in the environment, and loss of ability to complete the mission.

SC-3: The UAS must be capable of completing the mission objectives [H-3].

This system constraint is the reciprocal of H-3, and it must be enforced to prevent the potential loss of ability to complete the mission.

3.3 Developing the H ierarchal Control Model

3.3.1 Overview of STPA Step Two

The second step of STPA is to develop a hierarchical, functional control structure of the system required for the hazard analysis. A hierarchical control structure is a system model that is composed of feedback control loops like the one shown in Figure 2. Every control structure includes controllers, control actions, feedback, other inputs to and from components, and a controlled process. The hazard analysis is performed on the interactions between the components in the feedback-control loop.

In general, a *controller* may provide *control actions* to control some processes and to enforce constraints on the behavior of the *controlled process*. The *control algorithm* represents the controller's decision-making process—how it determines what control actions to implement and when to implement them. Controllers also have *process models* that represent the beliefs about the process being controlled, the environment, or other aspects of the system. The process models may be updated in part by *feedback* from the controlled process. For humans, the process model is generally referred to as a mental model and the control algorithm is the operating procedures, process, or rules employed to make decisions.

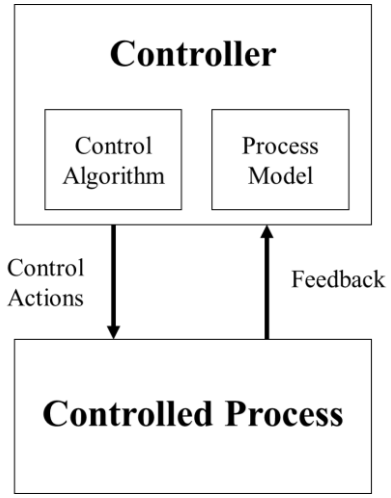

Figure 2. Generic Control Structure [3]

The hierarchical nature of the control structure is an important aspect of the model. Each controller that is above another in the model has authority over other controllers or controlled processes immediately below it. All downward arrows represent control actions or commands, and the upward arrows represent feedback to the controllers. Inputs and outputs are neither control actions nor feedback but are additional communication required between controllers that are necessary to enforce safety constraints on the system.

Abstraction is used throughout the development of the control structure to assist in managing the complexity of the system and to proceed with analysis before the entire

system is developed. The principle of abstraction can be applied to the controllers and controlled processes; the control actions and feedback paths in the control structure; and the specific mechanisms by which the controller acts upon a controlled process and mechanisms by which the controller receives feedback. Rather than explicitly modeling every detail of the system some parts of the system are abstracted while still maintaining the important interactions among the components.

The control structure can be developed through a systematic process. Each one of these components has responsibilities to ensure that together the system safety constraints are enforced. To develop the control structure, the responsibilities of each of the components are identified. Then, the control actions and feedback required to fulfill the responsibilities and enforce the system constraints can be determined.

3.2.2 Hierarchical Control Model for Human Supervisory Control of Multi-UAS

In this sub-section, the control model for human-supervisory control is presented. Figure 3 shows the model and a detailed description of the responsibilities, associated control actions and required feedback is discussed.

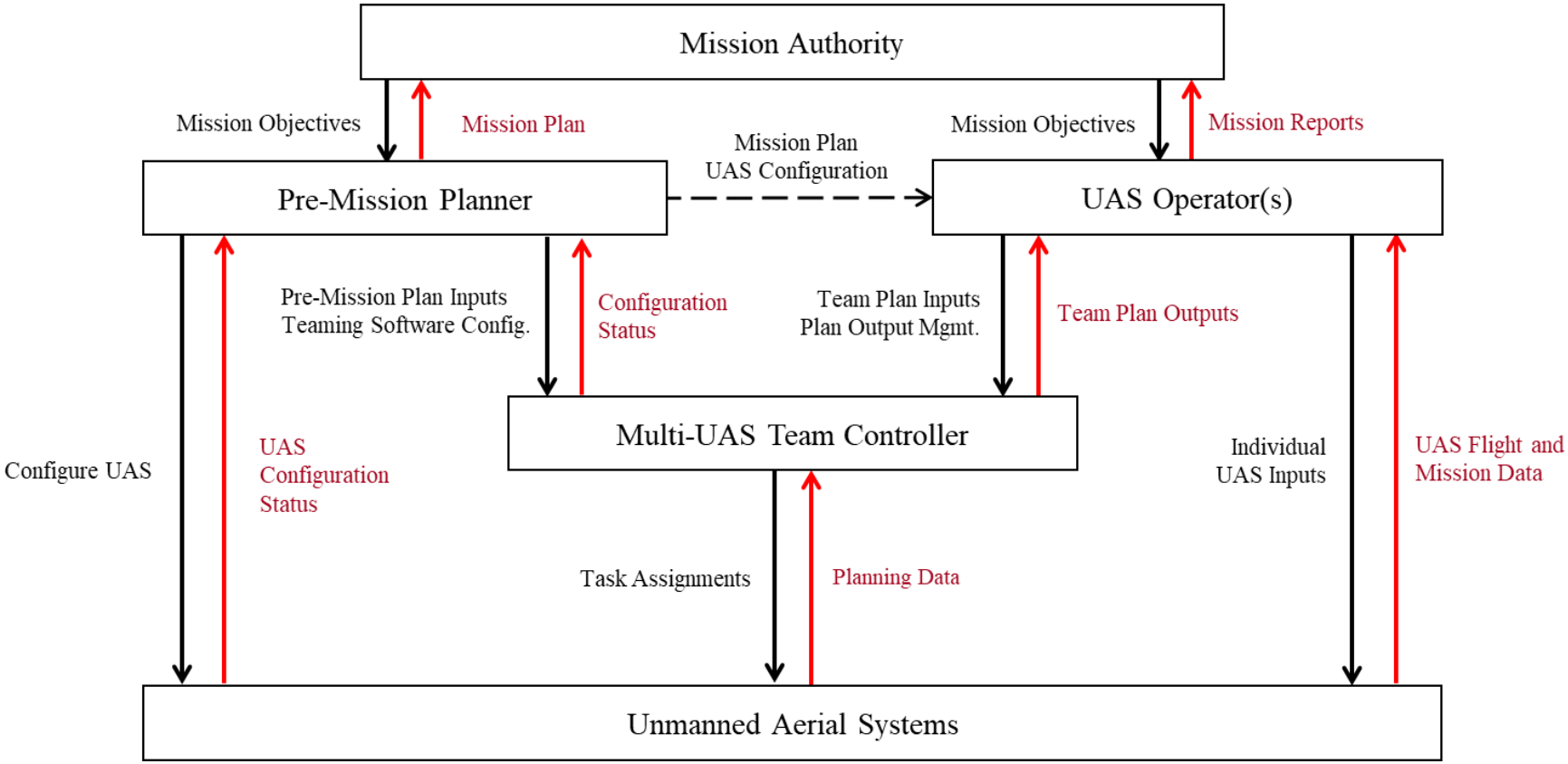

Figure 3. Hierarchical Control Model of Human Supervisory Control of Multi-UAS Systems

Mission Authority: The Mission Authority controller is a functional abstraction of the entities that are responsible for the definition and dissemination of the mission objectives and restrictions to those controlling the UAS. The Mission Authority provides the mission objectives before flight to the Pre-Mission Planners. These objectives ensure pre-mission activities, including pre-mission plan development, can be accomplished. The Mission Authority receives feedback on the pre-mission plans and provides pre-mission plan approval. During the mission, the Mission Authority provides any required updates to the mission objectives to the Operator. The Mission Authority maintains an accurate mental model of the mission through feedback provided from the Operator on the status of the mission objectives throughout the operation.

This description is formalized into the responsibilities, control actions, and feedback for the Mission Authority detailed below and represented in the control structure:

- **R-1:** Oversee the definition and dissemination of the mission objectives and restrictions [SC-3].

Table 3. Mission Authority Control Actions

Control Action	**Given to**	**Description**
Mission Objectives and Approval	Pre-Mission Planners	Abstraction of any guidance the Mission Authority provides before flight to the Pre-Mission Planners to define the objectives or restrictions for the mission [R-1].
Mission Objectives	Operator	Abstraction of any guidance the Mission Authority provides during flight to the Operator to adjust the objectives or restrictions for the mission [R-1].

Table 4. Mission Authority Feedback

Feedback	**Received from**	**Description**
Mission Plan	Pre-Mission Planners	Abstraction of any feedback on the mission plans developed from the objectives. This may include the feasibility of accomplishing the objectives given the resources.
Mission Status Reports	Operator	Abstraction of any feedback the Operator provides during flight on the status of the mission. It is assumed during this analysis that the Operator and Mission Authority are geographically separated. So, there will be some type of digital communication between them.

Pre-Mission Planner: The Pre-Mission Planner controller is a functional abstraction of any entities who are responsible for configuring the UAS and the Multi-UAS Team Controller before the mission. To fulfill this responsibility, they configure each UAS with the required hardware and software for the mission. For example, this may include selecting the appropriate payload or sensor configuration for each UAS based on the mission objectives or providing updates to the navigation software based on the mission location. The Pre-Mission Planner receives feedback on the successful configuration of each UAS through Pre-Mission inspection and testing.

The Pre-Mission Planner also ensures the Multi-UAS Team Controller has the required software to develop plans for the UAS team. This software may have to be updated periodically. For example, updates may be required because of flaws discovered in the planning software during previous missions. Successful configuration of the software by the Pre-Mission Planner requires an accurate mental model of how the Multi-UAS Team Controller will develop plans. This mental model is updated by feedback on the current software configuration and tests before the mission.

After the teaming software is configured, the Pre-Mission Planner oversees pre-mission plan development. They develop plans based on the mission objectives and provide initial planning inputs to the Multi-UAS Team Controller. For example, these pre-planning inputs may include a list of expected mission tasks, prioritization of these tasks, and other parameters such as no-fly zones. These inputs form the basis for the initial plans the UAS will follow. The Pre-Mission Planner receives feedback that the system is appropriately configured with the pre-mission inputs from the Multi-UAS Team Controller before the mission.

This description is formalized into the responsibilities, control actions, and feedback for the Pre-Mission Planner detailed below and represented in the control structure:

- **R-2**: Configure the UAS and Multi-UAS Team Controller before the mission [SC-1, SC-3].
- **R-3:** Develop a pre-mission plan and provide inputs required to configure the UAS and Multi-UAS Team Controller before flight [SC-3].

Table 5. Pre-Mission Planners Control Actions

Control Action	**Given to**	**Description**
Configure UAS	UAS	Abstraction of any of the required actions to configure each UAS before flight [R-2].
Teaming Control Software Configuration	Multi-UAS Team Controller	Abstraction of any of the actions required before flight to configure or update the software that allows for plan development during the mission [R-2].
Pre-Mission Plan Inputs	Multi-UAS Team Controller	Abstraction of any of the mission input parameters that may be provided to develop initial plans for the UAS. These inputs may vary based on the mission or multi-UAS application. However, this control action is purposefully abstracted to encapsulate a wide range of pre-mission inputs that may be provided [R-3].

Table 6. Pre-Mission Flight Planners Feedback

Feedback	**Received from**	**Description**
UAS Configuration	UAS	Abstraction of any feedback the Pre-Mission Planners receive on the configuration of the UAS from inspection or testing performed before flight.
Multi-UAS Team Controller Configuration	Multi-UAS Team Controller	Abstraction of any feedback the Pre-Mission Planners receive on the configuration of the Multi-UAS Team Controller. This may include feedback on the software used to develop plans and optimize UAS resources or confirmation of the pre-mission inputs.

UAS Operator(s): The UAS Operator controller is a functional abstraction of the human(s) who provide human-supervisory control over the UAS during operations. Consistent with the literature review, the Operator is responsible for oversight of the plan development, approval, and execution during the mission. To fulfill these responsibilities, the operator provides planning inputs to the Multi-UAS Team Controller during flight. These inputs may be revisions to the inputs previously provided by the Pre-Mission Planners to account for changes in the objectives or dynamic mission environment or new inputs that were not provided during pre-mission planning. As an example, the planning inputs provided during flight may consist of a list of prioritized mission tasks, mission restrictions, and any other planning parameters. For a military reconnaissance mission, these may include a list and prioritization of targets to investigate, a search area, and time restrictions to complete the mission within.

After the Multi-UAS Team Controller has developed a plan, the Operator reviews it and provides plan output management. This includes approving, denying, or modifying any aspect of the team plans. As reflected in the literature, the involvement of the Operator in providing plan output management may vary throughout the mission. The Operator may review and provide plan output management for every plan, or may only intervene if they believe that the developed plan will not accomplish the mission objectives. To effectively oversees plan development and execution, the Operator must have a mental model of the proposed plans, environment, and the state of each UAS. The Operator receives feedback from the Multi-UAS Team Controller on the proposed plans and receives flight and mission data from each UAS to update their mental model of the environment and UAS state.

In addition to providing oversight for the plan development and execution, the Operator may also provide direct control over the flight, navigation, and payloads of a single UAS in an emergency. The Operator retains authority over the control of each UAS in the team. If the Multi-UAS Team Controller is unable to effectively control or develop plans for the UAS, the operator can provide direct control over a single or multiple UAS. The Operator has a control action directly to the UAS that bypasses the Multi-UAS Team Controller. To provide effective direct control over a UAS, the operator receives feedback on the flight and mission data from each UAS.

This description is formalized into the responsibilities, control actions, and feedback for the Operator detailed below and represented in the control structure:

- **R-4:** Supervise the UAS team plan development, approval, and execution during the mission [SC-1, SC-3]
- **R-5:** Provide direct control of UAS in situations where the UAS cannot be safely and effectively controlled by itself or the Multi-UAS Team Controller [SC- SC-2, SC-3].

Table 7. UAS Operator Control Actions

Control Action	**Given to**	**Description**
Team Plan Inputs	Multi-UAS Team Controller	Abstraction of any inputs the Operator may provide to translate the mission objectives into actionable planning parameters for the Multi-UAS Team Controller. These inputs may vary based on the mission or multi-UAS application. However, this control action is purposefully abstracted to encapsulate a wide range of inputs that may be provided [R-4].
Plan Output Management	Multi-UAS Team Controller	Abstraction of any actions that are required by the operator after the plan has been developed. This may include approving, denying, or modifying aspects of the plan. This control action is purposefully abstracted to account for systems employing management by exception, management by consent, or adaptive automation [R-4]
Individual UAS Inputs	UAS	Abstraction of any actions to directly control the flight, navigation, or payloads of a sub-set or a single UAS [R-5]

Table 8. UAS Operator Feedback

Feedback	**Received from**	**Description**
Team Plan Output	Multi-UAS Team Controller	Abstraction of any feedback the Operator receives of the plans developed by the Multi-UAS Team Controller. This may include both the plans and justification for the plan development.
UAS Flight and Mission Data	Multi-UAS Team Controller	Abstraction of any feedback the Operators receive on the status of the mission tasks, and flight data for each UAS. It is assumed that the Operator does not have direct visual sight of the UAS during operations. Therefore, this feedback does not involve physical observance of the UAS. The feedback is provided through some other form (e.g., video stream, written text updates, etc.). The specific means by which the Operator receives feedback is not distinguished in this analysis.

Multi-UAS Team Controller: The Multi-UAS Team Controller is a functional abstraction of the software responsible for dynamically optimizing UAS resources and developing task assignments to accomplish the mission. The Multi-UAS Team Controller develops and sends task assignments to control each UAS throughout the mission. To fulfill this responsibility the Multi-UAS Team Controller must have an accurate process model of the mission objectives, environment, and state of each UAS. To update its process model of the mission objectives they receive planning inputs parameters from the Operator. They may request additional inputs from the Operator in the event there have been insufficient planning parameters provided, or changes in mission require revision to the previously provided parameters. For the Multi-UAS Team Controller to maintain an accurate process model of the environment and UAS state, they receive feedback from each UAS with the required planning data. This may include the tasks currently allocated to a UAS, the status of the current task assignment, and the flight data for each UAS.

If a UAS cannot complete its assigned tasks, the Multi-UAS Team Controller may replan and distribute tasks to a UAS that can complete the tasks. In the event the Multi-UAS Team Controller cannot develop plans given the mission inputs and state of each UAS, they may escalate the problem for deconfliction to the Operator.

This description is formalized into the responsibilities, control actions, and feedback for the Operator are detailed below and represented in the control structure:

- **R-6:** Generate individual task allocations of the UAS team to accomplish the mission based on the mission planning inputs provided by the UAS operator(s) [SC-1, SC-2, SC-3]

Table 9. Multi-UAS Team Controller Control Actions

Control Action	**Given to**	**Description**
Task Assignments	UAS	Abstraction of all of the task assignments that the Multi-UAS Team Controller may give to the UAS. This control action is purposefully abstracted to account for a wide range of task assignments that may be given during the mission. This abstraction allows for the analysis and recommendation to apply to a wide range of multi-UAS systems. [R-4]

Table 10. Multi-UAS Team Controller Feedback

Feedback	**Received from**	**Description**
UAS Planning Data	UAS	Abstraction of any feedback the Multi-UAS Team Controller may require to develop plans for the UAS. This may include current task allocation for each UAS, the status of the task, and the flight and mission data for each UAS.

Unmanned Aerial Systems: The UAS(s) controlled process is the functional abstraction of all the UAS that are a part of the system. In this analysis, the team of UAS is considered a single controlled process. Each UAS is responsible for controlling its flight, navigation, and payloads to achieve the task objectives assigned to it. It uses onboard sensors to collect information in its environment required to accomplish safe and effective operations. The UAS provides onboard flight and mission data to the Multi-UAS Team Controller and Operators throughout the mission. This feedback may include a wide range of data including its task status, health or battery status, and location. In the event a UAS is unable to accomplish a task, it provides feedback to the Multi-UAS Team Controller.

This description is formalized into the responsibility of the UAS:

- **R-7:** Control the flight, navigation, and payloads of the UAS [SC-1, SC-2, SC-3].

3.2.3 Summary of Control Structure developed for Human-Supervisory control of Multi-UAS systems

In summary, this section presented an abstracted hierarchical control model for the human-supervisory control of multi-UAS systems. Abstraction was used in the development of the control structure to ensure the model was both representative of the key control characteristic identified in the literature review and applies to a wide range of UAS implementations. The model includes interactions between pre-mission personnel, Operators, UAS teaming automation, and the UAS. The next section of this chapter uses the model to examine how the interactions among the identified components may lead to a hazardous system state.

3.4 Identifying the Unsafe Control Actions

3.4.1 Overview of STPA Step Three

The third step of STPA is to identify the unsafe control actions (UCAs) that can occur within the control structure. Control actions become unsafe specifically because of the context in which they are executed. Therefore, each UCA includes information about the control action and the context of *when* it may lead to a hazardous system state in a worse-case environmental condition. There are four types of ways that a control actions can be unsafe [3]:

1. A control action required to avoid a hazard is not provided or followed.
2. An unsafe control action is provided that leads to a hazard.
3. A potentially safe control action is provided too early, too late, or out of sequence that leads to a hazard.
4. A safe control action is stopped too soon or applied too long (for continuous control actions, not discrete ones) and leads to a hazard.

An unsafe control action contains five parts. The first part is the controller that can provide the control action. The second part is the type of unsafe control action (provided, not provided, too early or too late, stopped too soon, or applied too long). The third part is the control action or command itself. The fourth part is the context in which the control action may become unsafe. The last part is a link to the hazardous system state that the UCA may cause. Below is an example of an unsafe control action developed for a pilot that unsafely provides a break command during takeoff:

UCA-X:	Pilot	provides	Brake command	during a normal takeoff	[H-4.3]
	<Source>	<Type>	<Control Action>	<Context>	<Hazard Link>

3.4.2 Unsafe Control Actions for Human Supervisory Control of Multi-UAS Systems

The analysis identified 29 high-level UCAs across the control structure. The full list of UCAs can be found in Appendix A. UCA O-1.1 is presented here as an example of when the Operator is *not providing* the Team Plan Inputs could lead to a hazard:

> **UCA O-1.1:** The Operator does not provide Team Plan Inputs to the Multi-UAS Team Controller when the mission objectives or conditions have changed and the UAS team plans need to be updated. As a result, no plan is developed, or a plan is developed that is based on missing inputs [H-3].

Table 11 details some of the other types of UCAs related to the Operator providing Team Planning Inputs, as well as UCAs related to Plan Output Management provided by the Operator. These examples highlight only some of the possible ways UCAs can

occur within control actions and do not cover every possible case for the system. Each UCA is tied directly to the hazardous system state that it could lead to denoted by [H-].

Table 11. Selected Operator Unsafe Control Actions

Control Action	**Not Providing**	**Providing**	**Too early / too late / wrong Sequence**	**Applied too long / stopped too short**
Operator to Multi-UAS Team Controller: **Team Planning Inputs**	**UCA O-1.1** ...when the mission objectives or conditions have changed and the UAS team plans need to be updated. [H-3].	**UCA O-1.2** …when the mission conditions have not changed, and the new inputs are inconsistent with the mission objectives. [H-3].	**UCA O-1.3** …before the mission objectives have been confirmed. [H-3].	**UCA O-1.4** …stopped before all inputs required to develop a plan have been provided [H-3].
Operator to Multi-UAS Team Controller: **Plan Output Management (Approve, Deny or Modify Plan)**	**UCA O-2.1.** …when the Multi- UAS Team Controller has developed a plan that meets the mission objectives, and a confirmation is required to execute the mission. [H-3].	**UCA O-2.2** …when the plan developed by the Multi-UAS Team Controller does not meet the mission objectives. [H-3].	**UCA O-2.4** … provides too late and the plan developed by the Multi-UAS Team Controller are no longer achievable. [H-3].	

3.5 Generating the Causal Loss Scenarios

3.5.1 Overview of STPA Step Four

The fourth step of STPA is to identify causal loss scenarios (CS) that describe the causal factors of *why* an unsafe control action may occur, ultimately leading to the specified hazards and losses. In STPA, scenarios include more than just a single event or component failures. They include the context of how a combination of factors can lead to a hazard by breakdowns in the feedback control loop. The specific context in each scenario allows for detailed design recommendations to be made for the system.

Several approaches can be used to generate causal loss scenarios, and it is important to note that it is a creative process that cannot be reduced to a checklist. The STPA handbook describes one way to analyze each UCA by considering breakdowns that can occur throughout the feedback control loop. These are described below and Figures 4 and 5 show where the breakdown can occur in the control structure:

1. **Unsafe Controller Behavior:** The controller receives adequate feedback, but still provides (or does not provide) a control action that leads to a hazard.
2. **Inadequate Feedback Path and Other Inputs**: The controller does not receive adequate feedback, resulting in an unsafe control action decision by the controller
3. **Unsafe Control Path**: The controller provides a safe control action, but it is improperly received or is not received at all by the controlled process
4. **Unsafe Controlled Process Behavior**: The controlled process receives a safe control action, but it is improperly executed or is not executed at all.

It is important to note that UCAs can occur because of breakdowns in multiple parts of the feedback control loop, and these are not mutually exclusive and should not be considered as such. This is partially indicated by the overlap in the circles in Figures 4 and 5. In fact, causal loss scenarios often involve breakdowns in multiple parts of the control structure. These four breakdowns in the control structure are only a starting point for scenario generation. Once a starting point in the control structure is determined it is possible to consider how other factors in the control structure or environment may also play a contributing role.

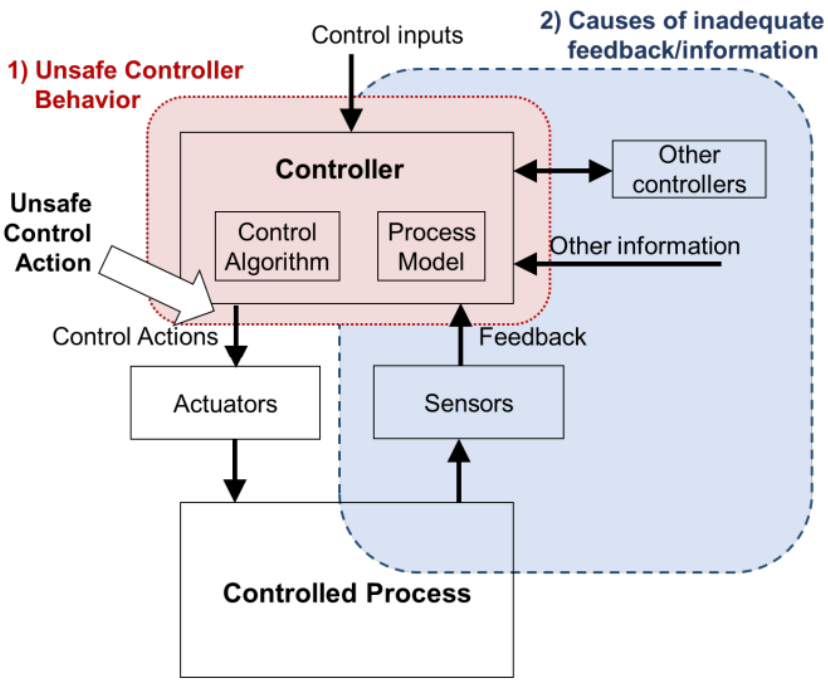

Figure 4. Breakdowns related to Unsafe Controller Behavior & Unsafe Feedback Path [3]

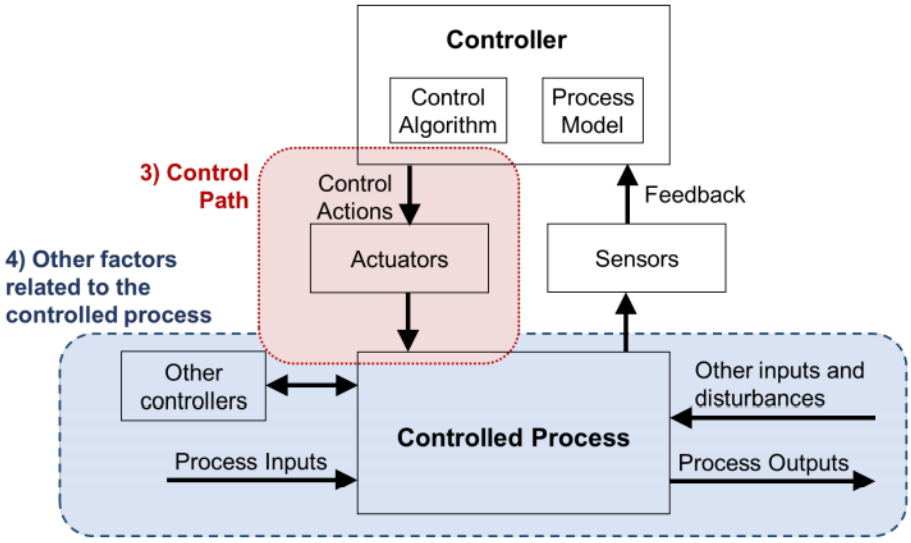

Figure 5. Breakdowns related to Unsafe Control Path & Unsafe Process Behavior [3]

3.5.2 Causal Loss Scenario for Human Supervisory Control of Multi-UAS Systems

In this analysis, 40 high-level scenarios were identified and refined into a total of 130 more detailed scenarios. The full list of scenarios is presented in Appendix B. In this section, 4 scenario examples are presented for UCA O-1.1. The scenarios presented in this section show how a UCA can be caused by breakdowns throughout the control structure including unsafe controller behavior [CS-1]; unsafe feedback [CS-2]; unsafe control path [CS-3]; and unsafe process behavior [CS-4].

UCA O-1.1: The Operator does not provide Team Planning Inputs to the Multi-UAS Team Controller when the team plans need to be updated to accomplish the mission. As a result, the system enters a state where the mission cannot be accomplished [H-3].

Scenario involving Unsafe Controller Behavior

CS-1: The Operator may not provide the Team Planning Inputs *because they have an incorrect mental model that the necessary inputs have already been pre-programmed by the Pre-Mission Planners*. As a result, either no plans are formulated, or a plan is developed based on only a subset of the required information. There are several reasons why the Operator may have an incorrect mental model of the inputs already provided.

> CS-1.1: There was never a clear delineation of responsibility established and discussed before the mission on what inputs the Pre-Mission Planner and Operator would each provide. Both assumed the other would provide the inputs for the mission. The Operator was unaware that there were missing inputs because the system did not allow them to review inputs provided (or not provided) by other users.
>
> CS-1.2: The Operator believed that the required inputs were already provided by the Pre-Mission Planner before the mission. However, the input provided by the Pre-Mission Planner was based on a previous version of mission plans provided by the Higher Mission Authority and these inputs were never updated to reflect the most up-to-date plans. However, this change was never communicated to the Operator.

CS-1.1 and CS-1.2 are examples of scenarios where unsafe controller behavior occurs because the controller's process model does not match reality because of improper coordination between multiple controllers.

Scenario involving a Breakdown in the Feedback Path

CS-2 The Operator may not provide the Team Planning Inputs *because the feedback is not salient because of disturbances in the Operator's environment*. As a result, either no plans are formulated. Feedback may be missing to inform the Operator that inputs are required because:

> CS-2.1: The Multi-UAS Team Controller is unaware that the feedback was not received because the system is not designed such that the Operator must acknowledge the feedback was successfully received.

CS-2.1 is an example of a scenario that involves the breakdown in the feedback path—the feedback is sent by the controlled process but is not received by the controller. The problem is not identified because the system is not designed to provide feedback on whether the feedback is received by the controller.

Scenario related to an Unsafe Control Path

CS-3: The Operator may provide the Team Planning Inputs, but the Multi-UAS Team Controller never receives inputs *because a disturbance in the environment (e.g., signal jamming) interrupted the successful transmission of the inputs.* As a result, a plan is never developed or a plan is developed based on partial inputs. This may be unrecoverable if:

> CS-3.1: The system is not designed to provide feedback to the Operator on whether inputs have been successfully received by the Multi-UAS Team Controller. So, the Operator does not take the necessary action to provide additional inputs. As a result, no plans are developed for the new mission.

CS-3.1 involves multiple portions of the feedback control structure. It begins with a breakdown in the control path where the control action is sent by the controller but not received by the controlled process. However, missing feedback because of the design of the system means the Operator is unable to make a correct decision to take necessary action to provide the inputs again. As a result, the Multi-UAS Team Controller does not develop a plan for the mission.

Scenario related to Unsafe Process Behavior Path

CS-4: The Operator may provide the Team Planning Inputs, but *a failure in the Multi-UAS Team Controller causes the system to interpret old inputs as new inputs*. This could occur for several reasons:

> CS-4.1: The Multi-UAS Team Controller temporarily resets because a component failure leads to the system overheating. During the system reset the most recent inputs were not stored and upon restarting old inputs were entered into the system.

CS-4.1 is an example of a scenario related to the controlled process where the controlled process does not respond properly.

3.5 Summary of Hazard Analysis of Human-Supervisory Control of Multi-UAS Systems

This chapter presented an overview of an STPA hazard analysis performed on the human-supervisory control of Multi-UAS systems. The analysis began by identifying undesirable system losses that framed the purpose and scope of the analysis. These losses included (L-1) loss of mission, (L-2) loss of life or permanently disabling injury, and (L-3) loss or damage to UAS or equipment. From these losses, system-level hazards and constraints were derived. The next step of the analysis presented an abstracted hierarchical control model for the human-supervisory control of multi-UAS systems. Abstraction was used in the development of the control structure to ensure it was both representative of the key control characteristic identified in the literature review and applies to a wide range of UAS implementations. The rest of the analysis focused on systematically determining when and why the system constraints may not be properly enforced in the system, ultimately leading to hazardous and losses. The results of the hazard analysis are used in the following chapter to provide design recommendations for the human-supervisory control of multi-UAS systems.

Chapter 4

Implications of STPA Hazard Analysis for the Human-Supervisory Control of Multi-UAS Systems

4.1 Chapter Overview

This chapter demonstrates how the results of the STPA hazard analysis were used to provide insights for the early design phases for Multi-UAS systems with human supervisory control. Each scenario generated in the STPA hazard analysis provides information to develop recommendations for the design and operation of the system. Recommendations were developed to enforce constraints on the system that eliminate the occurrence of the scenario altogether, or ensure measures are in place to mitigate the impact if the scenario cannot be fully eliminated.

Overall, 80 design recommendations were generated from the STPA hazard analysis to help ensure safe and effective multi-UAS operations. Recommendations were provided across the entire operation from pre-planning to in-flight control of the system. The recommendations address causal loss scenarios that arose from unsafe interactions between the UAS, automation to control the UAS, personnel controlling the UAS during operations, and others who influence the system, such as those who configure the system before the mission or provide guidance on the objectives of the mission.

The following sections highlight several of the design recommendations that were systematically developed from the STPA hazard analysis. A full list of design recommendations is provided in Appendix B. In each of the following sections, a UCA and scenario are provided which is reflective of a theme that arose in the generation of the scenarios and recommendations. Some examples of recommendations related to the theme are provided, as well as links to additional recommendations related to the theme. This organization was inspired by [45].

4.2 Highlights of Recommendations Generated from Analysis

4.2.1 Recommendations to Improve Handoff between Multiple Controllers

UCA O-1.1: Operator does not provide Team Planning Inputs when the mission objectives or conditions have changed and the UAS team plans need to be updated. As a result, the system enters a state where the mission cannot be accomplished [H-3].

CS-5: The Operator may not provide the Team Planning Inputs *because they incorrectly believe that the necessary inputs have already been pre-programmed by the Pre-Mission Planners*. As a result, either no plans are formulated, or a plan is developed based on only a subset of the required information.

> CS-5.1: There was never a clear delineation of responsibility established and discussed before the mission on what inputs each would provide. Both assumed the other would provide the inputs (e.g., no-fly zones requirements) for the mission. The Operator was unaware that there were missing inputs because the system did not allow them to review what inputs had been provided by other users [UCA O-1.1].

In multi-UAS systems, the responsibility of providing UAS team planning inputs will likely be distributed between multiple users including Pre-Mission planners and multiple operators during the mission. This distribution of responsibility has the potential to reduce the workload required of a single operator during flight. However, UCA O-1.1 and CS-1.1 highlight one example of potential missing control when multiple controllers are responsible for providing control over a single process. It may be especially difficult to discover gaps in required control inputs as represented in CS-5.1 when (a) the inputs are provided over a long-time span (i.e., days or hours between mission planning and execution), (b) when those involved are geographically separated or have limited communication, or (c) the design of the system makes it difficult to verify what inputs have already been received. From UCA O-1.1, CS-5.1, and other related scenarios, several design recommendations were developed to improve the coordination of multiple users in providing team planning inputs. A subset of these recommendations is presented in Table 12, and additional recommendations related to handoff between multiple users can be found in Appendix B associated with UCA O-2.1.

Table 12. Selected Recommendations related to Handoff between Multiple Controllers

	Design Recommendation
DR-1	There must be a clear delineation of responsibility for every person that provides inputs to the system. These responsibilities must be documented and communicated prior to and during the mission [UCA, O-1.1, CS-1.1].
DR-2	The design of the system must allow users to determine what planning inputs have been provided by themselves and other users [UCA O-1.1, CS-1.1]
DR-3	The Operators must receive feedback when their inputs have been altered by another user [UCA O-1.1, CS-7]

4.2.2 Recommendations to Improve Feedback for Human-Machine Trust Development

UCA O-2.1: Operator does not provide *Approve Plan* when the Multi-UAS Team Controller has developed a plan that meets the mission objectives, and the Multi-UAS Team Controller requires confirmation to execute the mission tasks. As a result, the system enters a state where the mission cannot be accomplished [H-3].

CS-14: The Operator may not Approve the plan *because there is no way to independently verify or receive feedback that the plan developed by the Multi-UAS Team Controller is proper for the mission objectives.* So, they do not approve the plan out of fear that it is incorrect, or mistrust of the automation [UCA O-2.1].

The operation of multi-UAS systems will require a paradigm shift from operators providing direct control over a single UAS to operators working with an autonomous controller to develop and execute plans for the entire team. One challenge that arose in the hazard analysis in several causal loss scenarios was ensuring that operators have the required feedback to effectively review plans developed by the Multi-UAS Team Controller. If the operators cannot understand the rationale of why the plan was developed, then they may be ineffective at reviewing the plans to ensure they are consistent with the mission objectives. Operators must receive proper feedback to build an appropriate level of trust in the autonomous controller's ability to develop plans. UCA O-2.1 and CS-14 highlight one example where the operator's lack of ability to verify the plan and distrust in the autonomous controller led to disapproval of a proper plan. From UCA O-2.1, CS-14, and others related to it, several design recommendations were developed related to the required feedback for proper human-machine trust development and plan review. A subset of these recommendations is presented in Table 13, and additional recommendations related to proper plan approval can be found in Appendix B associated with UCA O-2.2.

Table 13. Selected Recommendations related to Human-Machine Trust Development

Design Recommendation	
DR-4	Users must have the ability to independently verify the plans developed by the Multi-UAS Team Controller [UCA O-1.1, CS-1.1]
DR-5	The Operator must receive feedback on the rationale for the plan development, including the ability to query why parts of the plan were developed [UCA O-1.1, CS-1.1]
DR-6	The Operator must receive feedback if a UAS task assignment is determinantal to a single-UAS, but is required for the execution of the mission. [UCA O-3.3, CS-28].
DR-7	The feedback the Operator receives of the plan must be presented and organized in such a way that it assists the Operator in maintaining an accurate mental model of the proposed plan [UCA O-2.1, CS-13]

4.2.3 Recommendations to Assist the Operator in Providing Timely Plan Approval

UCA O-2.4: Operator provides *Approve Plans* too late and the plan developed by the Multi-UAS Team Controller is no longer achievable. As a result, the system enters a state where the mission cannot be accomplished [H-3].

CS-20: The Operator approves plans too late because they did not receive feedback that plan approval was time-sensitive. They do not receive feedback that the plan approval was time-sensitive because:

> **CS-20.1** No feedback was designed into the system to alert the Operator to time-sensitive plan approvals [UCA O-2.4].
>
> **CS-20.2:** The request for plan approval when originally sent to the Operator was not time critical. However, the plan request became time-sensitive because of changes in the environment (e.g., movement of the target in reconnaissance mission) and the system is not designed to adjust the feedback provided to the Operator [UCA O-2.4].

Multi-UAS missions are dynamic, and changes in the mission objectives, UAS state, or the environment will require a continual response from the operators to provide timely inputs. UCA O-2.4 and CS-20 highlight the concern where a change in the mission environment requires an operator to provide inputs within a restricted time window. However, feedback was not provided to the operator to alert them to the change in time sensitivity. If the Operator is not updated to the change in required inputs this could lead to an inability to execute the mission objectives. From UCA O-2.4, CS-20, and others related to it, several design recommendations were developed to ensure operators have the necessary feedback to provide time-critical responses. A subset of these recommendations is presented in Table 14, and additional requirements related to providing timely inputs can be found in Appendix B for UCA O-2.4, CS-12 through CS-16.

Table 14. Selected Recommendations related to Timely Plan Approval

Design Recommendation	
DR-8	The Operator must receive feedback when they are required to provide plan output management [UCA O-2.1, CS-15].
DR-9	The Operator must receive feedback that indicates when a plan is time-sensitive [UCA O-2.1, CS-20.1].
DR-10	The Operator must receive feedback when a non-time-sensitive plan approval becomes time-sensitive [UCA O-2.1, CS-20.2].
DR-11	The Operator must receive feedback during flight when a component failure has reduced the ability of the system to provide feedback on required plan output management [UCA O-2.1, CS-15].
DR-12	Components of the feedback system must be checked during pre-flight inspection [UCA O-2.1, CS-15].

4.2.4 Recommendations to Improve Operator to Machine Input Semantics

UCA O-1.1: The Operator does not provide Team Planning Inputs to the Multi-UAS Team Controller when the mission objectives or conditions have changed and the UAS team plans need to be updated. As a result, the system enters a state where the mission cannot be accomplished [H-3].

CS-2: The Operator may not provide the team planning inputs *because the team input parameters or semantics are inadequate to translate the Operator's desired intent for the team into machine-actionable inputs.* This scenario can be refined into several reasons why the operator may be unable to provide the required inputs:

> CS-2.2: The input mechanism only has a pre-loaded set of options for the operator to select from (set by the Pre-Mission Planners) and these options are not indicative of the inputs the Operator requires.
>
> CS-2.3 The Operator's privileges and responsibility to give inputs is limited, and these limits preclude them from providing the necessary inputs. A specific example is if there are multiple Operators, and it is assumed that one Operator will always be responsible for providing a certain set of inputs. So, the other Operator(s) is not given the authority or clearance to provide the required inputs.

The delegation of the UAS team plan development to the Multi-UAS Team Controller has the potential to allow operators to oversee more complex operations. However, operators are now faced with the challenge of translating their desired intent into actionable machine inputs. The increased mission complexity requires operators to provide a rich set of input considerations (including who, what, when, where, and how) to plan the team. In dynamic mission environments this may not always be simple, and UCA O-1.1, CS-2 demonstrates this challenge. This change may require exploration to understand what will be required in the system semantics so the operator can specify the more complex missions. In addition, it may also likely require alternative training and skills development for future operators that must be considered. From UCA O-1.1, CS-2 several design recommendations were developed to improve the ability of Operators to provide planning input parameters. A subset of these recommendations is presented in Table 15, and additional requirements related to translating mission intent into machine-actionable inputs can be found in Appendix B associated with UCA O-3.1, CS-24.

Table 15. Selected Recommendations related to Human-Machine Input Semantics

Design Recommendations	
DR-13	The user input mechanism must allow the Operator to translate their intent into actional machine inputs. This includes all mission variables the Operator may be required to provide during the mission [UCA O-1.1, CS-2.1]
DR-14	If there are a pre-loaded set of options for the Operator to select from, they must include all options required during flight. Pre-flight inspection must include a test that all required mission inputs will be available to the Operator [UCA O-1.1, CS-2.2]
DR-15	Some authority be responsible for ensuring operators have the appropriate level of permission to provide inputs prior to flight [UCA O-1.1, CS-2.3].
DR-16	The Operator must have a mechanism to request additional input permissions in the event they do not have them during flight [UCA O-1.1, CS-2.3].

4.2.5 Recommendations to Improve UAS Task Assignments Generation

UCA A-1.1: Multi-UAS Team Controller does not provide Task Assignments to UAS when the UAS(s) are ready to execute proper mission tasks that have been approved. As a result, the system enters a state where the mission cannot be accomplished [H-3]

> CS-29: The Multi-UAS Team Controller does not provide task assignments because it does not receive feedback that the UAS had completed its previous task [UCA A-1.1].
>
> CS-30: The Multi-UAS Team Controller provides task assignments, but the UAS never receives the task assignment because there is a disruption in the control path (e.g., UAS temporarily out of range). The Multi-UAS Team Controller does not receive feedback that tasks have not been successfully received by each UAS. So, they do not resend the task assignments [UCA A-1.1].
>
> CS-31: The Multi-UAS Team Controller provides task assignments, but the UAS can no longer perform the task. This feedback to alert the Multi-UAS Team Controller is either missing or has been disrupted. So, the Multi-UAS Team Control is unaware that they need to develop and provide new task assignments [UCA A-1.1].

The responsibility of the Multi-UAS Team Controller to develop plans and provide task assignments to each UAS requires accurate feedback on the state of each UAS. UCA A-1.1 and CS 29-31 are examples where the Multi-UAS Team Controller is unable to effectively develop plans and generate task assignments for the team because of breakdowns in the feedback-control path. From UCA O-1.1, CS-1, and other related scenarios, several design recommendations were developed to improve the ability of the Multi-UAS Team Controller to develop plans and generate task assignments. A subset of these recommendations is presented in Table 16, and additional requirements related to task generation can be found in Appendix B associated with UCA A-1.2.

Table 16. Selected Recommendations related to UAS Task Assignment Generation

Design Recommendations	
DR-17	The Multi-UAS Team Controller must receive feedback when a UAS has completed an assigned task [UCA A-1.1, CS-29].
DR-18	The Multi-UAS Team Controller must receive feedback on the transmission status of the task assignments [UCA A-1.1, CS-31].
DR-19	If the UAS does not receive the task assignments, the Multi-UAS Team Controller must provide the task assignment again [UCA A-1.1, CS-30].
DR-20	The Multi-UAS Team Controller must receive feedback when a UAS is incapable of performing a task [UCA A-1.1, CS-31].
DR-21	If a UAS can no longer perform a task, the Multi-UAS Team Controller must develop and provide new task assignments [UCA A-1.1, CS-31].

4.3 Summary of Design Recommendations Generated from Hazard Analysis

This analysis systematically generated design requirements from the STPA hazard analysis for the human supervisory control of multi-UAS systems. In the preceding section, several highlights of the design recommendations were presented and organized into themes that emerged from the scenarios and subsequent recommendations. These themes included recommendations to improve the handoff between multiple controllers, timely plan review and approval by the Operator, necessary feedback required for human-machine trust development, overcoming inadequate input semantics, and ability of the Multi-UAS Team Controller to generate task assignments plans. In addition to design recommendations, additional questions were raised that must be addressed to develop these systems.

Appendix B provides a full list of recommendations related to the scenarios that were generated for this analysis. Overall, recommendations were provided across the entire operation from pre-planning to in-flight control of the system. The recommendations address causal loss scenarios that arise from unsafe interactions between the UAS, automation to control the UAS, personnel controlling the UAS during operations, and others who influence the system, such as those who configure the system before the mission or provide guidance on the objectives of the mission.

Chapter 5

Thesis Summary

5.1 Research Contributions

To enable the benefits of multi-UAS operations requires a paradigm shift from multiple operators remotely *controlling* a single-UAS to a single operator *supervising* multiple-UAS with the assistance of autonomous controllers. To ensure operators and autonomous controllers can work together to safely control multi-UAS requires a rigorous safety-guided design process to inform design requirements in the early stages of system development. The objective of this thesis was to demonstrate how early safety-guided design recommendations can be developed that (1) enable safety to be designed into the system early when most effective, and (2) apply to a wide range of multi-UAS systems with human supervisory control.

To begin to address this shortfall, this thesis systematically generated design requirements from an STPA hazard analysis for the human supervisory control of multi-UAS systems, satisfying the thesis objectives. The STPA hazard analysis presented in this thesis began by identifying undesirable system losses including (L-1) loss of mission, (L-2) loss of life or permanently disabling injury, and (L-3) loss or damage to UAS or equipment. From these losses, system-level hazards and constraints were derived. The next step of the analysis presented an abstracted hierarchical control model for the human-supervisory control of multi-UAS systems. Abstraction was used in the development of the control structure to ensure it was both representative of the key control characteristic identified in the literature review and applies to a wide range of UAS implementations. The rest of the analysis focused on systematically determining when and why the system constraints may not be properly enforced in the system. The results of the hazard analysis were used to systematically generate design requirements for the human supervisory control of multi-UAS systems, fulfilling the objective of this thesis. These recommendations, if applied, can help ensure safety is built into the system from the early design phases.

5.2 Future Work

The work presented in this thesis could be expanded upon in several ways. First, as with any hazard analysis, the unsafe control actions and causal loss scenario generated from the control structure in this thesis is almost surely incomplete. Multi-UAS systems with human supervisory control would benefit from the generation of additional unsafe

control actions, causal loss scenarios, and design recommendations derived from the hierarchical control structure developed in this thesis. Second, STPA can be used as a tool to inform iterative safety-guided design whereby design recommendations are generated from the analysis, implemented in the control structure, and reevaluated with additional cycles of hazard analysis. The recommendations generated in this thesis could be used to refine the control structure, and iterative cycles of STPA hazard analysis could be performed until recommendations are provided for detailed system components. Third, to provide recommendations for components not considered in this analysis the scope of the system boundary could be expanded to include Air Traffic Controller, other aircraft in the environment, and others.

References

[1] J. Svan, “Pilot Error, Poor Visibility caused Lakenheath Pilot to crash in North Sea,” *Stars and Stripes*, 2020. https://www.stripes.com/news/europe/pilot-error-poor-visibility-caused-lakenheath-pilot-to-crash-in-north-sea-1.653097 (accessed May 01, 2021).

[2] N. Leveson, *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge: The MIT Press, 2011.

[3] N. Leveson and J. Thomas, *STPA Handbook*. Cambridge, 2018.

[4] T. Porat, T. Oron-Gilad, M. Rottem-Hovev, and J. Silbiger, “Supervising and controlling unmanned systems: A multi-phase study with subject matter experts,” *Front. Psychol.*, vol. 7, no. MAY, 2016, doi: 10.3389/fpsyg.2016.00568.

[5] “Pentagon Unmanned Systems Integrated Roadmap 2017-2042,” *USNI News*. https://news.usni.org/2018/08/30/pentagon-unmanned-systems-integrated-roadmap-2017-2042.

[6] C. M. Belcastro, R. L. Newman, J. K. Evans, D. H. Klyde, L. C. Barr, and E. Ancel, “Hazards identification and analysis for unmanned aircraft system operations,” *17th AIAA Aviat. Technol. Integr. Oper. Conf. 2017*, no. June, 2017, doi: 10.2514/6.2017-3269.

[7] G. Skorobogatov, C. Barrado, and E. Salamí, “Multiple UAV Systems: A Survey,” *Unmanned Syst.*, pp. 149–169, 2020.

[8] S. Folse, “Systems-Theoretic Process Analysis of Small Unmanned Aerial System Use at Edwards Air Force Base,” Massachusetts Institute of Technology, 2017.

[9] J. Robertson, “Systems Theoretic Process Analysis Applied to Manned-Unmanned Teaming,” Massachusetts Institute of Technology, 2019.

[10] Concepción de León, “Drone Delivery? Amazon Moves Closer With F.A.A. Approval,” *The New York Times*, 2020. https://www.nytimes.com/2020/08/31/business/amazon-drone-delivery.html.

[11] N. Francesco and F. Remondino, “UAV for 3D mapping applications: A Review,” *Appl. Geomatics*, vol. 6, no. 1, 2014.

[12] G. Bevacqua, J. Cacace, A. Finzi, and V. Lippiello, “Mixed-Initiative Planning and Execution for Multiple Drones in Search and Rescue Missions,” 2015.

[13] D. Camara, “Cavalry to the rescue: Drones fleet to help rescuers operations over disasters scenarios,” 2014.

[14] V. Sharma, H.-C. Chen, and R. Kumar, “Driver Behaviour Detection and Vehicle Rating using Multi-UAV Coordinated Vehicular Networks,” *J. Comput. Syst. Sci.*, vol. 86, pp. 3–32, 2017.

[15] M. Abdelkader, M. Shaqura, C. G. Claudel, and W. Gueaieb, “A UAV Based System for Real Time Flash Flood Monitoring in Desert Environments using Lagrangian Microsensor,” in *IEEE 2013 Int. Conf. Unmanned Aircraft Systems (ICUAS)*, 2013, pp. 25–34.

[16] P. Tripicchio, M. Satler, G. Dabisias, E. Ruffaldi, and C. A. Avizzano, “Towards Smart Farming and Sustainable Agriculture with Drones," 2015 International Conference on Intelligent Environments,” 2015.

[17] A. Kopeikin *et al.*, “Designing and Flight-Testing a Swarm of Small UAS to Assist

Post-Nuclear Blast Forensics," 2020.
[18] A. Kopeikin, S. Heider, D. Larkin, C. Korpela, R. Morales, and J. E. Bluman, "Unmanned Aircraft System Swarm for Radiological and Imagery Data Collection," 2019.
[19] J. Y. C. Chen, M. J. Barne, and M. Harper-Sciarini, "Supervisory Control of Multiple Robots: Human-Performance Issues andUser-Interface Design," *IEEE Trans. Syst. Man. Cybern.*, vol. 41, no. 2, 2011.
[20] M. Cummings and S. Guerlain, "Developing operator capacity estimates for supervisory control of autonomous vehicles," *Hum. Factors Jounral Hum. Factors Ergon. Soc.*, vol. 49, no. 1, pp. 1–15, 2007.
[21] A. Saif, Osamah; Fantoni, Isabelle; Zavala-Río, "Distributed integral control of multiple UAVs: Precise flocking and navigation," *IET Control Theory Appl.*, vol. 13, no. 13, pp. 2008–2017, 2019, doi: 10.1049/iet-cta.2018.5684.
[22] A. L. P. Mattei, E. S. A. Orbital, C. F. M. Toledo, J. da S. Arantes, and O. T. Jr., "Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019.
[23] S. Ramchurn, J. Fischer, Y. Ikuno, F. Wu, J. Flann, and A. Waldock, "A Study of Human-Agent Collaboration for Multi-UAV Task Allocation in Dynamic Environments," 2015.
[24] M. Cummings, A. Clare, and C. Hart, "The Role of Human-Automation Consensus in Multiple Unmanned Vehicle Scheduling," *Hum. Factors Jounral Hum. Factors Ergon. Soc.*, vol. 52, no. 1, pp. 17–27, 2010.
[25] S. van Lochem, C. Borst, G. C. H. E. de Croon, M. M. van Paassen, and M. Mulder, "Ecological Interface for Collaboration of Multiple UAVs in Remote Areas," *IFAC-PapersOnLine*, vol. 49, no. 19, pp. 450–455, 2016, doi: 10.1016/j.ifacol.2016.10.620.
[26] G. Algan, "Development of a dynamic flying digital display based on autonomous swarm of UAVs," Technishe Universiteit Eindhoven, 2014.
[27] J. Scherer *et al.*, "An Autonomous Multi-UAV System for Search and Rescue," in *DroNet '15: Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, 2015, pp. 33–38.
[28] B. Steele, "Intel's latest light show was the first FAA-approved drone swarm," *Engadget*, 2016. https://www.engadget.com/2016-05-05-intel-faa-approved-drone-swarm-light-show.html.
[29] J. Hu *et al.*, "To centralize or not to centralize: A tale of swarm coordination.," 2018.
[30] H. A. Ruff, G. L. Calhoun, M. H. Draper, and J. V. Fontejon, "Exploring Automation Issues in Supervisory Contol of Multiple UAVs," in *Proceedings of the Human Performance, Situation Awareness, and Automation Technology Conference*, 2004, pp. 218–222.
[31] M. Cummings and P. J. Mitchell, "Managing Multiple UAVs Through a Timeline Display," *Am. Inst. Aeronaut. Astronaut.*, 2005, doi: 10.2514/6.2005-7060.
[32] W. Olson and N. Sarter, "Automation Management Strategies: Pilot Preferences and Operational Experiences," *Int. J. Aviat. Psychol.*, vol. 10, no. 4, pp. 327–341, 2000.
[33] W. Olson and M. Sarter, "Management by Consent in Human-Machine Systems:

When and Why it Breaks Down," *Hum. factors J. Hum. Factors Soc.*, vol. 43, no. 2, 2001.
[34] H. A. Ruff, S. Narayanana, and M. H. Draper, "Human Interaction with Levels of Automation and Decision-Aid Fidelity in the Supervisory Control of Multiple Simulated Unmanned Air Vehicles," *Presence Teleoperators Virtual Environ.*, vol. 11, no. 4, pp. 335–351, 2002.
[35] R. Parasuraman, E. de Visser, and K. Cosenzo, "Adaptive Automation for Human Supervision of Multiple Uninhabited Vehicles: Effects on Change Detection, Situation Awareness, and Mental Workload," *Mil. Psychol.*, vol. 12, no. 2, pp. 270–297, 2009.
[36] G. L. Calhoun, C. Miller, M. H. Draper, and H. A. Ruff, "Adaptable Automation Interface for Multi-Unmanned Aerial Systems Control," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 2013, pp. 26–30.
[37] B. Donmez, C. Nehme, and M. Cummings, "Modeling Workload Impact in Multiple Unmanned Vehicle Supervisory Control," *IEEE Trans. Syst. Man Cybern. - Part A Syst. Humans*, vol. 40, no. 6, pp. 1180–1190, 2010.
[38] C. Aerospace, "Unmanned Aerial Systems," 2021. https://www.collinsaerospace.com/what-we-do/military-and-defense/platforms/unmanned-air.
[39] H. Fang and M. Duan, "Safety System Engineering for Offshore Oil," in *Offshore Operation Facilities*, Gulf Professional Publishing, 2014, pp. e183–e347.
[40] C. Fleming and N. Leveson, "Including safety during early development phases of future air traffic management concepts," 2014.
[41] N. Leveson, *Safeware: System Safety and Computers*. Addison-Wesley, 1995.
[42] S. Kabir, "An overview of fault tree analysis and its application in model based dependability analysis," *Expert Syst. Appl.*, vol. 77, pp. 114–135, 2017.
[43] S. Summers, "Systems Theoretic Process Analysis Applied to Air Force Acquisition Technical Requirements Development," Massachusetts Institute of Technology, 2018.
[44] D. Horney, "Systems-Theoretic Analysis and Safety-Guided Design of Military Systems," Massachusetts Institute of Technology, 2017.
[45] A. Kopeikin *et al.*, "Rotary-Wing Aircraft Development," Cambridge, 2021.

Appendix A: STPA Unsafe Control Actions

Unsafe Control Actions - UAS Operator

Control Action	Not Providing	Providing	Too early / Late	Applied too long / Stopped too short
Operator to Multi-UAS Team Controller: Team Planning Inputs	**UCA O-1.1** Operator does not provide *Team Planning Inputs* when inputs are required to develop a plan that meets the accounts for changes in mission conditions. As a result, the system enters a state where the mission cannot be accomplished [H-3].	**UCA O-1.2** Operator provides *Team Planning Inputs* when the new inputs are inconsistent with the mission objectives. As a result, the system enters a state where the mission cannot be accomplished [H-3].	**UCA O-1.3** Operator provides *Team Planning Inputs* before the Multi-Team Controller can accept inputs.	**UCA O-1.4** Operator provides only a portion of the entire *Team Planning Inputs* required when all inputs are required to develop a plan that is consistent with the mission objectives. As a result, the system enters a state where the mission cannot be fully accomplished [H-3].
Operator to Multi-UAS Team Controller: Plan Output Management (Approve, Deny or Modify Plan)	**UCA O-2.1.** Operator does not provide *Approve Plan* when the Multi-UAS Team Controller has developed a plan that meets the mission objectives and the Team Controller requires confirmation to execute the mission tasks. As a result,	**UCA O-2.2** Operator provides *Approve Plans* when the plan developed by the Multi-UAS Team Controller does not meet the mission objectives. As a result, the system enters a state where the mission cannot be accomplished [H-3].	**UCA O-2.3** Operator provides *Approve Plans* too early before the plan that will meet the mission objectives has been fully developed by the Team Controller. As a result, the system enters a state where the mission	

	the system enters a state where the mission cannot be accomplished [H-3].		cannot be accomplished [H-3]. **UCA O-2.4** Operator provides *Approve Plans* too late and the plan developed by the Multi-UAS Team Controller is no longer achievable. As a result, the system enters a state where the mission cannot be accomplished [H-3]. **UCA O-2.5** Operator provides Approve Plans out of order during a coordinated sequence of plans that must be accomplished in a particular order As a result, the system enters a state where the mission cannot be accomplished [H-3].	
Operator to Multi-UAS Team Controller: Individual UAS Inputs	**UCA O-3.1** Operator does not provide Individual Plan Inputs when the Multi-UAS Team Controller assigns tasks to UAS that do not meet the mission objective and the	**UCA O-3.3** Operator provides Individual Plan Inputs when the UAS is executing a proper task assignment issued by the Team Controller. As a result, the		

	Operator is unable to provide an intervention through the Team Controller. As a result, the system enters a state where the mission cannot be accomplished; the UAS is uncontrollable; or the UAS may violate minimum or maximum separation requirements. [H-1, H-2, H-3] **UCA O-3.2** Operator does not provide Individual Plan Inputs when the UAS does not follow the assignments provided by the Team Control and the operator cannot control the UAS through the Multi-UAS Team Controller. As a result, the system enters a state where the mission cannot be accomplished; the UAS is uncontrollable; or the UAS may violate minimum or maximum separation requirements. [H-1, H-2, H-3]	system enters a state where the mission cannot be accomplished; the UAS is uncontrollable; or the UAS may violate minimum or maximum separation requirements. [H-1, H-2, H-3]		

Unsafe Control Actions – Multi-UAS Team Controller

Control Action	Not Providing	Providing	Too early / Late	Applied too long / Stopped too short
Multi-UAS Team Controller to UAS: Task Assignment	**UCA A-1.1** Multi-UAS Team Controller does not provide *Task Assignments* to UAS when the UAS(s) are ready to execute proper mission tasks that have been approved. As a result, the system enters a state where the mission cannot be accomplished [H-3]	**UCA A-1.2** Multi-UAS Team Controller provides *Task Assignments* to UAS before the plan has been approved by the operator. As a result, the system enters a state where the mission cannot be accomplished [H-3]	**UCA A-1.3** Multi-UAS Team Controller provides new *Task Assignment* before the UAS has completed the previous mission tasks that are required to accomplish the mission. As a result, the system enters a state where the mission cannot be accomplished [H-3] **UCA A-1.4** Multi-UAS Team Controller provides Task Assignment too late after the proper mission task is no longer achievable. As a result, the system enters a state where the mission cannot be accomplished [H-3]	**UCA A-1.5** Multi-UAS Team Controller provides only a portion of the entire *Task Assignment* when all of the task assignments must be assigned to complete the mission. As a result, the system enters a state where the mission cannot be accomplished [H-3]

Unsafe Control Actions – Mission Authority

Control Action	Not Providing	Providing	Too early / Late	Applied too long / Stopped too short
Mission Authority to UAS Operator: Mission Objectives	**UCA M-1.1.** Mission Authority does not provide *Mission Objectives* when they have changed. As a result, the system enters a state where the mission cannot be accomplished [H-3]		**UCA M-1.2** Mission Authority provides *Mission Objectives* too late and the new objectives are now unachievable. As a result, the system enters a state where the mission cannot be accomplished [H-3]	
Mission Authority to Pre-Mission Planning and Maintenance: Mission Objectives	**UCA M-1.1.** Mission Authority does not provide *Mission Objectives* when they are required for Pre-Mission planning [H-3]		**UCA M-1.1.** Mission Authority does not provide *Mission Objectives* when they have changed. As a result, the system enters a state where the mission cannot be accomplished [H-3]	

Unsafe Control Actions – Pre-Mission Planner

Control Action	Not Providing	Providing	Too early / Late	Applied too long / Stopped too short
Pre-Mission Planning and Maintenance to UAS: Configure UAS	**UCA P-1.1** The Pre-Mission Planning Team does not *Configure UAS* when they must be configured before the mission. As a result, the system enters a state where the mission cannot be accomplished; the UAS is uncontrollable; or the UAS may violate minimum or maximum separation requirements. [H-1, H-2, H-3]	**UCA P-1.2** The Pre-Mission Planning Team *Configure UAS* incorrectly before the mission. As a result, the system enters a state where the mission cannot be accomplished; the UAS is uncontrollable; or the UAS may violate minimum or maximum separation requirements. [H-1, H-2, H-3]		
Pre-Mission Planning and Maintenance to Multi-UAS Team Controller: Pre-Mission Plan Inputs	**UCA P-2.1** The Pre-Mission Planning Team does not provide *Pre-Mission Plan* Inputs to the Multi-UAS Team Controller when the inputs are required to develop and approve a plan before the mission. As a result, the system enters a state where	**UCA P-2.1** The Pre-Mission Planning Team provides the incorrect *Pre-Mission Plan Inputs to* the Multi-UAS Team Controller before the mission, and an incorrect pre-mission plan is developed. As a result, the system enters a state where		

	the mission cannot be accomplished [H-1]	the mission cannot be accomplished [H-1].		
Pre-Mission Planning and Maintenance to Multi-UAS Team Controller: Updates Team Optimization Software or Algorithm	**UCA P-3.1** The Pre-Mission Planning Team does not provide *Updates to the Team Optimization Software or Algorithm* when the current software or optimization algorithm will not allow proper plan development during the mission. As a result, the system enters a state where the mission cannot be accomplished [H-1]	**UCA P-3.1** The Pre-Mission Planning Team provides *Updates to the Team Optimization Software or Algorithm* when the current settings will accomplish the mission and any changes will not. As a result, the system enters a state where the mission cannot be accomplished [H-1]		

Appendix B: Causal Loss Scenarios & Recommendations

This appendix presents examples scenarios for a sub-set of the Unsafe Control Actions identified in Appendix A. The scenarios presented in this section are not exhaustive. However, the scenarios that are presented provide a basis for initial requirements and future hazard analyzes to reference and build upon. The author attempted to provide scenarios across the entire system, instead of providing scenarios for UCAs given by one or a limited number of controllers in the system.

Scenarios for Unsafe Control Actions provided by the Operator

Control Action: Team Planning Inputs

UCA O-1.1: Operator does not provide *Team Planning Inputs* to Multi-UAS Team Controller when the mission objectives or conditions have changed, and the team plans need to be updated. As a result, the system enters a state where the mission cannot be accomplished [H-1].

CS-1: The Operator may not provide new inputs because they are unaware the mission objectives have changed because they were not provided by the Mission Authority. Why could there be missing or inadequate feedback:

- The communication between the Mission Authority and Operator is disrupted by objects in the surroundings, or (for a military mission) enemy communication jamming.

Design Recommendation derived from CS-1:

- The Operator must receive feedback from the Mission Authority when the mission objectives have changed. If there is a disruption in the feedback path the operator must be notified [UCA O-1.1].
- The Mission Authority must have an alternative means to communicate changes in mission objectives in the event the primary communication channel is unavailable.

CS-2: The Operator may not provide inputs because the mechanism by which the Operator translates the mission objectives into inputs for the Team Control is limited in a way that will not allow the Operator to input what is required. Why might the mechanism not allow required inputs?

- The input mechanism only has a pre-loaded set of options for the operator to select from loaded by the Pre-Mission Planners. These options are not indicative of the entire mission space required to execute the mission.

Design Recommendation derived from CS-2:

- The input mechanism must allow the Operator to translate the mission objectives into actionable machine inputs.
- If there are pre-loaded set of options for the Operator to select from, they must include all options required during flight. Communication must occur between the Pre-Mission Planners and Operators to review the expected inputs prior to flight.

CS-3: This could occur if the Operator is overwhelmed or occupied by other tasks that must be completed and is physically or mentally incapacitated. The Operator may be overwhelmed because of other responsibilities including:

- Providing control over other systems (e.g., the operator is operating a manned vehicle)
- Reporting to Mission Authority
- The Operator has a number of different tasks to complete, and the system does not assist them in managing which tasks to accomplish first

Design Recommendation derived from CS-3:

- The system must assist the operator in determining which inputs are time critical and executing providing them in an appropriate time.

CS-4: This could occur if the Operator believes they have already provided the required updates when they have not. The Operator may have an incorrect mental model that inputs have been provided because:

- The Operator makes a lapse in memory and does not provide the inputs but believes they have already provided them. There is no mechanism to alert the Operator when inputs have been received. So, the Operator believes they have successfully provided inputs when they were in fact never provided.
- The Operator provided inputs, but they were never received by the Multi-UAS Team Controller. The Operator did not receive feedback that the inputs were not received by the Multi-UAS Team Controller. So, they did not take the proper action to provide additional inputs.

Design Recommendation derived from CS-4:

- The Operator must receive salient feedback until they have provided the required inputs.

- The Operator must receive feedback on the inputs they have previously provided.
- The Operator must receive feedback when the inputs have been successfully received by the Multi-UAS Team Controller.

CS-5: This could occur where there is more than one Operator responsible for providing the required inputs. The operator believes a different Operator is responsible for providing inputs when they are not. There might be a disconnect between responsibilities if:

- There is a communication breakdown between the operators, or there is not a clear line of responsibility set between the operators.

Design Recommendations derived from CS-5:

- There must be a clear delineation of responsibility on what inputs each Operator will provide before and during the mission.
- The system must assist the Operators in determining what inputs have been provided by other operators.

CS-6: This could occur if the mission objectives are changing at a pace where the Operator cannot translate the objectives into inputs before the inputs become outdated.

Design Recommendation derived from CS-6:

- The system must assist the Operator in providing timely inputs. This may involve allowing the Operator to provide different levels of inputs that can be refined so that the input process must not begin anew in the event of a minor change in the mission environment.

CS-7: This could occur if the Operator provides the correct inputs, but they are altered by another Operator (non-malicious, authorized Operator or non-authorized, malicious third party) who provides conflicting or inconsistent inputs. The Operator is unaware they have been altered and is unable to take corrective action.

Design Recommendations derived from CS-7:

- The Operator must receive feedback when changes have been made to the inputs they provided.
- The Operator must receive information on the source of other inputs to discern if a malicious actor has gained access to the system.

<u>UCA O-1.2:</u> Operator provides *Team Planning Inputs* to Multi-UAS Team Controller when the mission conditions have not changed and the new inputs are inconsistent with the mission objectives. As a result, the system enters a state where the mission cannot be accomplished [H-3].

CS-8: The Operator provides inputs and the Multi-UAS Team Controller begins plan development. However, the Multi-UAS Team controller is awaiting feedback from a UAS to complete planning and it is taking longer than normal to develop a plan. The Operator does not receive feedback that the Multi-UAS Team Controller is in the process of developing a plan. So, they believe the Multi-UAS Team Controller did not receive the inputs, or is unable to develop a plan given the inputs. The Operator provides the inputs again, which restarts the plan development. This creates a perpetual cycle where planning cannot be completed.

<u>Design Recommendation derived from CS-8:</u>

- The Operator must receive feedback when the Multi-UAS Team Controller is in the process of developing a plan.

CS-9: The Operator provides incorrect inputs because they an incorrect mental model of the mission objectives because they only received partial feedback on the mission objectives from the Mission Authority. This could occur if the communication between the Mission Authority and Operator is disrupted by objects in the surrounding, or (for a military mission) enemy communication jamming.

<u>Design Recommendation derived from CS-9:</u>

- Consistent with recommendation provided to address CS-1.

CS-10: The Operator performs an error in execution and unknowingly provided Team Planning Inputs when they did not intend to do so. They may either be (a) unaware they provided the inputs, or (b) aware they provided the inputs and are unable to fix the error. As a result, the Multi-UAS Team Controller receives improper inputs. The Operator may unintentionally provide inputs because:

- The Operator had an incorrect mental model that they were providing inputs only to a single UAS because the system did not assist the operator in distinguishing the two different inputs types. As a result, the plan inputs meant for a single UAS were provided for all the UAS.
- The Operator had an incorrect mental model that they were planning, or testing out future inputs, not current inputs. This is because the system did not assist the operator in distinguishing between initial inputs and finalized inputs.

There may be no way to overcome the slip if:

- The Operator does not receive feedback on the inputs they have provided.

- The Operator is unable to alter or revise the inputs once provided.

Design Recommendation derived from CS-10:

- The Operator must receive feedback to determine if they are providing inputs to a single UAS or for the entire UAS team.
- The system must assist the Operator in distinguishing between initial and finalized inputs.
- The Operator must receive feedback on the inputs they have provided.
- The Operator must have the ability to alter or revise the inputs once they have been provided.

CS-11: The Operator does not provide Team Plan Inputs, but the Multi-UAS Team Controller receives inputs that are inconsistent with the mission objectives because they were provided by another Operator.

Design Recommendation derived from CS-11:

- Consistent with design recommendations provided for CS-7.

Control Action: Plan Output Management (Approve, Deny, or Modify Plans)

UCA O-2.1: Operator does not provide *Approve Plan* when the Multi-UAS Team Controller has developed a plan that meets the mission objectives and the Multi-UAS Team Controller requires confirmation to execute the mission tasks. As a result, the system enters a state where the mission cannot be accomplished [H-3].

CS-12: The Operator does not approve the plans because they have an incorrect mental model that the plan is improper for the mission objectives because the feedback path was disrupted and they only receive part of the plan, or an outdated plan

Design Recommendations derived from CS-12:

- The system must ensure the Operator receives the full plan from the Multi-UAS Team Controller

CS-13 The Operator does not approve the plans because they have an incorrect mental model that the plan is improper for the mission objectives because the plan is very detailed and presented in such a way that the Operator cannot maintain an accurate mental model of the plan.

Design Recommendations derived from CS-13:

- The feedback the Operator receives of the plan must be presented and organized in such a way that it assists the Operator in maintaining an accurate mental model of the proposed plan

CS-14: The Operator does not approve the plan because they do not have the ability to independently verify that the plan is proper for the mission objectives. So, they do not approve the plan out of mistrust of the automation.

Design Recommendations derived from CS-14:

- The Operator must have a mechanism to independently verify the plan meets the mission objectives

CS-15 The Operator does not approve the plan because there was insufficient feedback to alert them that a plan must be approved for the mission to proceed. There may be insufficient feedback because:

- No feedback mechanism was incorporated into the design to alert the operator when they need to review / approve a plan.

- The feedback is only available for a certain period of time and the operator does not recognize that a plan must be approved before the feedback ends.
- Disturbances in the environment (noises, vibration, etc.) reduce the ability of the operator to perceive the feedback
- The feedback when working is adequate, but was not working as designed because of a component failure in the feedback system, or an attempt by a malicious actor to degrade the feedback

Design Recommendations derived from CS-15:

- The operator must receive feedback when they are required to provide plan output management.
- The feedback must persist until the Operator has provided plan output management.
- The feedback must be salient to include operator' environments that are loud or have significant vibrational disturbances. This may involve the use of dual feedback mechanisms in the event a single mode of feedback is ineffective at alerting the operator.
- Feedback components must be checked during pre-flight. The Operator must receive feedback during flight when a component failure has reduced the ability of the system to provide feedback on required plan output management.

CS-16: The Operator does not approve the plan because they have an incorrect mental model that they have already approved the approval. The Operator silences the feedback indicating that they have not provided inputs, and the design of the system does not assist the Operator in determining what plans have already been approved.

Design Recommendations derived from CS-16:

- The Operator must receive feedback on what plans they have and have not approved.

CS-17: The Operator does not approve the plans because they are physically incapacitated and there is no means to route commands to alternative operators or to alert the Mission Authority.

Design Recommendations derived from CS-17:

- The system must have a mechanism to alert other Operators in the event an Operator is incapacitated and cannot provide plan output management.

CS-18: The Operator approves the plan, but the plan approval is not sent to the Multi-UAS Team Controller because the control path is degraded (e.g., control path is saturated (bandwidth), obstructions in the environment interrupt the control path). The Operator is unaware the Multi-UAS Team Controller has not received the plans because no additional feedback is provided to the Operator.

Design Recommendations derived from CS-18:

- The Operator must receive feedback on the successful transmission of the inputs to the Multi-UAS Team Controller
- After [TBD] time the Multi-UAS Team Controller must send another request for plan approval to the Operator.

UCA O-2.2: Operator provides *Approve Plans* when the plan developed by the Multi-UAS Team Controller does not meet the mission objectives and the Multi-UAS Team Controller requires confirmation to execute the mission tasks. As a result, the system enters a state where the mission cannot be accomplished [H-3].

CS-19: The Operator may approve the plan because they have a flawed mental model of the plan based on the feedback they receive of the plan. They believe the plan will achieve the mission objectives when it will not. This could occur because:

- The Operator only receives feedback on the overview of the plan. From this feedback, the Operator receives the plan appears to meet the mission objectives. However, the detailed mission plans will not meet the mission objectives.
- The plan information is provided at a very detailed level. There is so much information that the operator cannot determine if the details of the plan (as a collective) will accomplish the mission.

Design Recommendations derived from CS-19:

- Similar to the recommendations provided for CS-13
- The Operator must have the ability to alter the level of feedback provided of the plan. It must allow them to discern an overview of the plan and analyze the details of the plan.

CS-20: The Operator slip (right intention, wrong execution) and approve a plan that they intended not to approve. The initial slip could occur because:

- Difficult to distinguish between plan approval and disapproval in the actuator mechanism
- Disturbances in the environment (vibration, noise, etc.) cause the Operator to accidently provide plan approval

There may no way to overcome the initial slip if the operator has no way to rescind previous plan approval once they realize they approved when they intended to not approve.

Design Recommendations derived from CS-20:

- Similar to the recommendations provided for CS-15
- The Operator must have a means to recall plan that have already been approved. More generally, they must have the ability to alter plan approvals within [TBD] seconds before the plans are executed.

UCA O-2.4: Operator provides *Approve Plans* too late and the plan developed by the Multi-UAS Team Controller is no longer achievable. As a result, the system enters a state where the mission cannot be accomplished [H-3].

CS-20: The Operator approves plans to late because they did not receive feedback that plan approval was time sensitive. They may have improper feedback because:

- No feedback was designed into the system to alert the Operator to time sensitive plan approvals
- The request for plan approval when originally sent to the Operator was not non-time critical. However, the plan request became time critical because of changes in the environment (e.g., movement of target in reconnaissance mission) and the system is not designed to adjust the feedback provided to the Operator.

Design Recommendations derived from CS-20:

- The Operator must receive feedback that indicated a plan is time sensitive
- The Operator must receive feedback when a non-time sensitive plan approval becomes time sensitive.

CS-21: This could occur if the feedback the Multi-UAS Team Controller receives from each UAS changes so rapidly that the plans change so frequently that the Operator cannot adequately review the plan before it becomes outdated.

Design Recommendations derived from CS-21:

-

CS-22: The Operator provides timely inputs. However, a disruption in the control path results in a [TBD] time delay between when the Operator approved the plans and the Multi-UAS Team Controller received the plans. As a result, the plans are out of date when received. The Multi-UAS Team Controller may either generate improper task assignments, or no task assignment are created.

Design Recommendations derived from CS-22:

- A secure control path must be established between the Operator and Multi-UAS Team Controller during plan approval.
- If there is a [TBD] time gap between when the plans are approved and when they are executed the Multi-UAS Team Controller must asses whether the plans can still be accomplished.

UCA O-2.5 Operator provides *Approve Plans* out of order during a coordinated sequence of plans that must be accomplished in a particular order. As a result, the system enters a state where the mission cannot be accomplished [H-3].

CS-23: This could occur if the system is designed such that the Operator can approve portions of the plan, instead of approving the entire plan. If the Operator proceeds to approve portions or phases of the plan out of order because they are unaware which phases of the plan must be completed first. This could occur if:

- The Multi-UAS Controller sends all of the plans in segments for the entire mission at once, and there is nothing in the design to assist the operator in determining which parts of the plan must be approved first.

Design Recommendations derived from CS-23:

- The feedback on the plans provided to the Operator must assist them in providing ordered plan approval.

CS-4: The Operator correctly approves tasks in order, but there is a temporary disruption in the control path between the Operator and the Multi-UAS Team Controller. If the control path is designed to send out individual parts of the plan without checking if previous parts of the plan were received then some parts of the plan will be sent out of order.

Design Recommendations derived from CS-24:

- A secure connection must be established between the Operator and Multi-UAS Team Controller.
- The system must ensure that all parts of the plan have been received by the Multi-UAS Team Controller in the correct sequence. If they are not properly received the Operator must be notified.

Control Action: Individual UAS Inputs

UCA O-3.1: Operator does not provide *Individual UAS Inputs* when the Multi-UAS Team Controller assigns tasks to UAS that do not meet the mission objective and the Operator is unable to provide an intervention through the Multi-UAS Team Controller. As a result, the system enters a state where the mission cannot be accomplished; the UAS is uncontrollable; or the UAS may violate minimum or maximum separation requirements. [H-1, H-2, H-3]

CS-24: The Operator is unable to provide individual UAS inputs because they do not have the authority to override the inputs provided by the Multi-UAS Team Controller.

Design Recommendations derived from CS-24:

- The Operator must have the ability to override the inputs provided by the Multi-UAS Team Controller

CS-25: The Operator does not provide individual UAS inputs because they are unable to determine why a UAS is performing a task. They trust the Multi-UAS Team Controller has provided correct task assignments when it has not.

Design Recommendations derived from CS-25:

- The Operator must receive feedback on the individual task assignments of each UAS.

UCA O-3.2: Operator does not provide *Individual UAS Inputs* when the UAS does not follow the assignments provided by the Team Control and the operator cannot control the UAS through the Multi-UAS Team Controller. As a result, the system enters a state where the mission cannot be accomplished; the UAS is uncontrollable; or the UAS may violate minimum or maximum separation requirements. [H-1, H-2, H-3]

CS-26: This could occur if the operator does not receive feedback that the individual UAS is not following the assignment because the system is in a state where the operator only receives feedback on high-level mission tasks.

Design Recommendations derived from CS-25:

- The Operator must receive feedback when the Multi-UAS Team Controller is unable to control the UAS.

CS-27: The Operator does not provide individual UAS inputs because they are overwhelmed by providing coordination for the entire UAS team. A loss will result if individual UAS inputs are not provided. However, they are unaware that it is a higher priority to provide direct control over a single UAS than coordinate the entire UAS.

Design Recommendations derived from CS-27:

- The Operator must receive feedback on the priority of providing individual UAS inputs versus team planning inputs.

UCA O-3.3: Operator provides *Individual UAS Inputs* when the UAS is executing a proper task assignment issued by the Multi-UAS Team Controller. As a result, the system enters a state where the mission cannot be accomplished; the UAS is uncontrollable; or the UAS may violate minimum or maximum separation requirements. [H-1, H-2, H-3]

CS-28: The Operator does not provide input because they receive feedback on a single UAS behavior and believes the assigned task assignment will lead to a loss (e.g., observes that a UAS is about to crash). However, the UAS task (even if detrimental to a single UAS) is required to achieve the more global mission objectives. The operator may not know that the loss of single UAS serves the global objective if the Operator does not receive feedback from the Multi-UAS Team Controller or UAS on why the UAS's seemingly detrimental action is required.

Design Recommendations derived from CS-28:

- The Operator must receive feedback if a UAS task assignment is determinantal to a single-UAS, but is required for the execution of the mission.
- The Operator must have the ability to request additional feedback on why a UAS was given a task assignment.

Scenarios for Unsafe Control Actions provided by the Team Controller

UCA A-1.1: Multi-UAS Team Controller does not provide *Task Assignments* to UAS when the UAS(s) are ready to execute proper mission tasks that have been approved. As a result, the system enters a state where the mission cannot be accomplished [H-3]

CS-29: The Multi-UAS Team Controller does not provide task assignments because it did not receive feedback that the UAS had completed its previous task.

Design Recommendations derived from CS-29:

- The Multi-UAS Team Controller must receive feedback when a UAS has completed an assigned task.

CS-30: The Multi-UAS Team Controller provides task assignments, but the UAS never receives the task assignment because there is a disruption in the control path (e.g., UAS temporarily out of range). The Multi-UAS Team Controller does not receive feedback that tasks have not been successfully received by each UAS. So, they do not resend the task assignments

Design Recommendations derived from CS-30:

- The Multi-UAS Team Controller must receive feedback on the transmission status of the task assignments.
- In the event that the UAS does not receive the task assignments, the Multi-UAS Team Controller must provide the task assignment again.

CS-31: The Multi-UAS Team Controller provides task assignments, but the UAS can no longer perform the task. This feedback is either missing or has been disrupted. So, the Multi-UAS Team Control is unaware that they need to develop and provide new task assignments.

Design Recommendations derived from CS-31:

- The Multi-UAS Team Controller must receive feedback when a UAS is incapable of performing a task.
- In the event that a UAS can no longer perform a task, the Multi-UAS Team Controller must develop and provide new task assignments.

CS-32: This could occur if the Multi-UAS Team Controller incorrectly send multiple tasks to a single UAS that should have been distributed to the UAS team because:

- Right before the tasks were sent there was a problem a UAS. The UAS was temporarily removed from the group because they had improper behavior. So, the tasks were sent to the available UAS, but some of the tasks were never assigned to the sub-set of UAS that were no longer part of the team.

Design Recommendations derived from CS-32:

- Correct task assignments must be provided to each UAS.
- Consistent with design recommendation provided from CS-30.

UCA A-1.2: Multi-UAS Team Controller provides *Task Assignments* before the plan has been approved by the operator. As a result, the system enters a state where the mission cannot be accomplished [H-3].

CS-33: This could occur if the mission changes slightly and the UAS is required to do (what it believes) is some minor task re-assignments. It believes these are within the bounds of the already approved plan. However, the new task assignments and slightly revised plan will actually have a larger impact on the mission than anticipated by the Multi-UAS Team Controller because the Operator has information that it does not. However, the new plan and task assignments are never sent to the Operator for re-approval.

Design Recommendations derived from CS-33:

- The Multi-UAS Team Controller must provide feedback to changes in the task assignments to the Operator
- The Operator must review the updated plan and provide plan output management when required

CS-34: This could occur if the Multi-UAS Team Controller has an incorrect process model that the operator is not required to approve the plan because the system believes the operator is incapacitated. So, it develops a plan and task assignment without the approval of the Operator. However, that feedback it receives on the operator is incorrect, and the Operator is in a capable state to review the plans.

Design Recommendations derived from CS-34:

- The Multi-UAS Team Controller must receive feedback on the state of the Operator.

Scenarios for Unsafe Control Actions provided by the Pre-Mission Planning Team

UCA P-1.2: The Pre-Mission Planning Team *Configure UAS* in a manner that is inconsistent with the mission objectives. As a result, the system enters a state where the mission cannot be accomplished; the UAS is uncontrollable; or the UAS may violate minimum or maximum separation requirements. [H-1, H-2, H-3]

CS-35: The Pre-Mission Planner properly configure each UAS based on the stated mission objectives. However, the mission objectives change immediately prior to flight and they do not receive feedback on the updated mission objectives. As a result, the UAS is not reconfigured.

Design Recommendations derived from CS-35:

- The Pre-Mission Planners must receive feedback on updated mission objectives.
- A final review process of each UAS must occur prior to flight

CS-36: The Pre-Mission Planners may configure UAS inconsistent with the mission objectives because there is no process to translate the mission objectives into the required configuration of each UAS.

Design Recommendations derived from CS-36:

- There must be a process to determine what each UAS will require based on the mission objectives.

CS-37: This could occur if the UAS are configured in more than one location. The coordination between multiple pre-mission planners is missing or interrupted. As a result, each believes the other has configured a particular UAS for the mission, but neither have.

Design Recommendations derived from CS-37:

- In systems where UAS are configured in separate geographical locations or by multiple Pre-Mission Planners, there must be coordination to ensure all UAS are properly configured.

CS-38: This could occur if the Pre-Mission Planning Team has no way to verify each UAS is properly configured because the mission environment or dynamics are too complex.

Design Recommendations derived from CS-38:

- The Pre-Mission Planner must have the ability to verify each UAS is properly configured for the mission.

UCA P-2.1: The Pre-Mission Planning Team provides the incorrect *Pre-Mission Plan Inputs to* the Multi-UAS Team Controller before the mission, and an incorrect pre-mission plan is developed. As a result, the system enters a state where the mission cannot be accomplished [H-1].

CS-39: This could occur if there is an incompatibility between the system accepting inputs before the mission and during the mission. For example, this could occur if the development of the pre-mission inputs is done on a simulator of the Multi-UAS Team Controller that will be used in the mission, not the real Controller. As a result, there is incompatibility between how the inputs are provided before the mission and how they are used during the mission.

Design Recommendations derived from CS-39:

- The Multi-UAS Team Controller simulator and flight system must be consistent.
- A review process of the inputs must be conducted prior to flight.

CS-40: This could occur if there is a vast difference between the stated objectives before the mission and what occurs during the mission. So, the inputs were correct at the time, but ended up being incorrect later on.

Design Recommendations derived from CS-40:

- Consistent with the recommendation provided for CS-5

UCA P-3.1: The Pre-Mission Planning Team does not provide *Updates to the Team Software* when the current software or optimization algorithm will not allow proper plan development during the mission. As a result, the system enters a state where the mission cannot be accomplished [H-1]

CS-40: This could occur if the Pre-Mission Planning Team has no way to verify or test that the current Multi-UAS Team Controller software or input will be ineffective because it will be used in a completely new environment or mission and there are no previous information to determine if it will work. Or, there is no testbed to determine how the algorithm or software will work before the mission. This scenario is also applicable to the UCA P3-2, except a change is made to the software when it would work properly, but there was no way to know if it would until the mission began.

Design Recommendations derived from CS-41:

- Testing must be conducted prior to flight that is representative of the required planning during the mission.
- The Pre-Mission Planner must have a process to determine if updates are required for the Multi-UAS Team software.

Scenarios for Unsafe Control Actions provided by the Higher-Mission Authority

UCA M-1.2: Mission Authority updates the Mission Objectives so often that the Operator and Multi-UAS cannot develop a plan before the objectives have changed. As a result, the system enters a state where the mission cannot be accomplished [H-3]

CS-42: This could occur if the Mission Authority receives real-time feedback of the mission, and seeks to control the mission in detail. So, instead of providing high-level guidance they start to give guidance on very specific aspects of the mission objectives. However, because they are not actually at the mission location, the feedback they receive is not complete. As a result, they are developing updated mission objectives that are not consistent with the overall goal of the mission.

Design Recommendations derived from CS-41:

- The Mission Authority must only provide high-level guidance on the objectives, and trust the Operator to translate the mission objectives into actionable team inputs.